Regional Gating Neural Networks for Multi-label Image Classification

Rui-Wei Zhao[1]
rwzhao14@fudan.edu.cn
Jianguo Li[2]
jianguo.li@intel.com
Yurong Chen[2]
yurong.chen@intel.com
Jia-Ming Liu[3]
james.liu.n1@gmail.com
Yu-Gang Jiang[1]
ygj@fudan.edu.cn
Xiangyang Xue[1]
xyxue@fudan.edu.cn

[1] Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China

[2] Intel Labs China Beijing, China

[3] Department of Control Science and Engineering Tongji University Shanghai, China

Abstract

This paper proposes a novel deep learning framework for multi-label image classification, namely regional gating neural networks (RGNN). The motivation is two folds. First, global image features (including CNN based features) ignore the underlying context information among different objects in an image. Consequently, people attempt to use information from objectness regions. However, current objectness region proposal algorithms usually produce several thousand region candidates, including many classification irrelevant or even noisy regions. This leads to the second problem: how to select useful contextual regions for image classification. RGNN is an end-to-end deep learning framework that can automatically select contextual region features with specially designed gate units, which are then fused for classification. Because the gate units and the classifier are integrated in the same deep neural network pipeline, we can learn parameters of the network simultaneously. We evaluate the proposed method on PASCAL VOC 2007/2012 and MS-COCO benchmarks, and results show that RGNN is superior to existing state-of-the-art methods.

1 Introduction

Image classification is a fast moving research area. Conventionally, handcrafted features like SIFT and traditional classifiers like SVM are utilized to classify the images. Recently, breakthrough has been achieved by deep convolutional neural networks (CNNs), which remarkably outperform traditional methods on several well-known image classification benchmarks like PASCAL VOC and ILSVRC [3, 13, 20].

The early deep learning based methods usually take size-normalized full images as inputs and feed them into a CNN network for classification. Since image classes are strongly related to the containing objects, the main limitation behind this framework is caused by its ignorance to the underlying context information among the objects in the images. This will easily cause the well-known semantic gap issue [22]. Another side effect is that size-normalization will change the aspect ratios of the input images, yielding various internal object distortions and thus harming the classification accuracy.

Region feature based methods have been proved to be helpful earlier than the prosperity of deep learning models [4, 9]. Recently, lots of proposed region proposal algorithms provide handy and economic solutions to produce comparatively high quality objectness region boxes on the input images [11]. Some existing works further suggest that coupling CNN with semantic region proposals can largely boost the image classification performance [24]. However, when applying region proposals directly to classification task, people observed that lots of irrelevant or even noisy (non-object) regions would definitely deteriorate the classification accuracy. Although some heuristic region selection/filtering procedures have been designed to handle this issue [15, 24, 25], they are far from optimal due to the lack of explicit objective function consistent to the classification task to guide contextual region selection.

In this paper, we propose a novel end-to-end deep learning framework named as regional gating neural networks (RGNN) to address the above limitations. RGNN takes raw images and a set of region proposals as inputs, and imposes gate units on regional deep representation to perform contextual region selection. Selected regional features are then fused for the final classification purpose. The gate units and classifier are integrated in one deep neural network, so all the parameters can be learnt together. Figure 1 illustrates the architecture of the proposed RGNN. To the best of our knowledge, we are the first to impose gate unit on CNN framework for contextual region selection and multi-label image classification with multi-task optimizations in an end-to-end manner. The major contributions of this paper are as follows

1. We propose RGNN, an end-to-end deep learning framework for multi-label image classification. RGNN can simultaneously select contextual regions with designed gate units, and perform classification over contextual image representations.

2. We study two different gate units: (1) region-level gate controls the pass/suppress for each region; (2) feature-level gate controls the pass/suppress for each feature dimension in every regions. One may choose either type of gate according to the practical needs.

3. We achieve state-of-the-art performances on PASCAL VOC 2007/2012 and MS-COCO multi-label classification benchmarks, solely based on the proposed method without multi-model fusion.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works and Section 3 elaborates the proposed RGNN. In Section 4, we conduct experiments to evaluate the proposed method, and conclusions are drawn in Section 5.

2 Related Works

Recently, there are many research efforts on combining advanced CNN models with semantic regions since image classes are strongly correlated to the objectness regions within

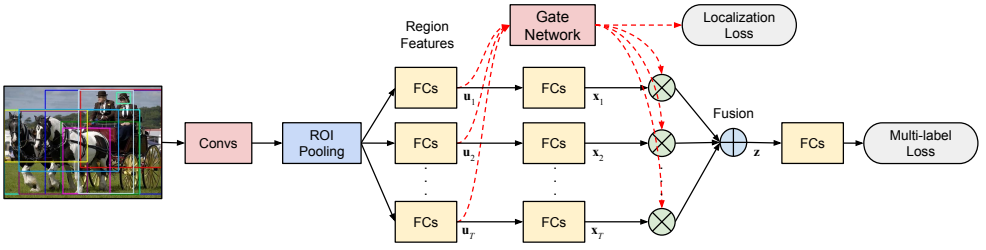


Figure 1: Architecture of RGNN. Layers such as pooling, ReLU and soft-max are omitted in the illustration for clarity. Example region proposals are depicted on the input image.

the image. Some early attempts like [8] make orderless pooling over all extracted CNN region features sampled at differently scaled sliding windows. For high quality candidate regions, many object proposals methods were proposed in the past few years, such as selective search [23], EdgeBox [28], BING [2], and so on. Based on these region proposals, Girshick et al. [6] invented RCNN to successfully combine CNN with local region information for object detection purpose, followed by the improved Fast RCNN [5] highlighting on improved efficiency in region features extraction. It is worth noting that Ren et al. [19] has also invented the Faster RCNN method for object detection, in which region bounding boxes are discovered by embedded region proposal networks rather than separated region proposal methods. We emphasize that different from [19], this work focuses mainly on the generic problem of optimal selecting contextual information given any kind of available image regions for multi-label classification purpose, regardless how these regions are generated.

When utilizing region proposals for the image classification purpose, region selection/ filtering is generally required [15, 16, 25]. This is because current region proposal algorithms usually produce a large portion of redundant or even noisy regions in order to achieve high objectness recalls. For instance, Wei et al. [24] proposed a CNN based method called Hypotheses-CNN-Pooling (HCP) which (1) removes some regions according to their sizes, aspect-ratios and confidence scores; (2) clusters regions into groups and keeps only one representative region in each group. Note that all of Luo et al. [15], Mettes et al. [16] and Wu et al. [25] made heuristic contextual region selection by using separate classifiers. The major limitations of these methods are due to the handcrafted rules or heuristic assumptions, which usually yield non-optimal solutions, especially for those with multi-stage training procedure. Compared to these existing methods, in this work we try to realize automatical region selection built in deep classification architectures and employ a special multi-task optimization scheme to help better network training.

Meanwhile, researchers already found that gate structures in LSTM [10] could be used to discover visual attentions and value instance/feature importance in some audio and visual tasks [18, 26]. Our designed gate units in RGNN framework are inspired by these works. However, we attempt to embed them seamlessly in the image classification framework to tackle the orderless and various numbered regions selection problem.

3 Our Approach

We propose RGNN to classify multi-label images, which realizes automatic contextual region feature selection with integrated gate units in an end-to-end learning framework. Figure 1 illustrates the RGNN architecture. The feed-forward path of RGNN consists of 5 steps:

1. *Region boxes*. For each image, object proposal method is applied to produce multiple candidate regions. We adopt EdgeBox for region proposal generation.

2. *Shared CNN networks*. Input images with proposal bounding boxes pass through a series of shared convolution/pooling layers. ROI pooling is then applied to every projected regions to obtain fixed size feature maps. FC layers are connected after ROI pooling layer to produce regional representation as vectors.

3. *Gate Units*. Gate units are imposed on each regional representation to control whether to be turned on/off so as to select useful contextual region features. In specific, two kinds of gates are described in Section 3.1, namely region-level gate and feature-level gate.

4. *Fused contextual representation*. Regional representation will be fused together to produce the contextual feature representation with a multi-scale cross region pooling layer. See more details in Section 3.2.

5. *Multi-label classification*. Fused contextual representation are fed into FC layers to predict image labels. The whole network is optimized with multi-label loss. When object level bounding box annotations are available, we further introduce a localization loss to optimize network by multi-task learning. See more details in Section 3.3.

3.1 Gate Units

Gate units are deployed to select contextual region features. The idea is inspired by the mechanism of gate unit in LSTM [10], which is used to learn to remember or forget the history information from long sequence of input data. Different from LSTM, our gate units do not depend on data at different time steps, but are elaborately designed to "remember" or "forget" features across different image regions.

Given an input image I, assume the region proposal algorithm extracts T regions denoted as $\mathcal{R} = \{R_1, \ldots, R_T\}$. The value of T may vary for different images. We formulate gate unit to produce fused contextual image representation $\mathbf{z}$ as

$$\mathbf{z} = f(\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_T),\ \hat{\mathbf{x}}_i = g(\mathbf{x}_i, \mathbf{u}_i) \tag{1}$$

where $\mathbf{x}_i$ is the original region features, which are transformed to $\hat{\mathbf{x}}_i$ by the gating function $g(\cdot)$ dependent on extra region information $\mathbf{u}_i$. And $f(\cdot)$ is the fusion function that combines multiple gated features into a contextual representation $\mathbf{z}$. In this paper, we design two kinds of gate structures named as region-level gate (Figure 2(a)) and feature-level gate (Figure 2(b)) respectively. They explore different contextual information among regions. People may choose either type of gate according to their practical needs.

Region-level gate controls the pass/suppress of each region features as a whole by their contribution to the classification. In other word, feature values within one region will be endowed with the same weight of importance. Figure 2(a) illustrates the architecture of

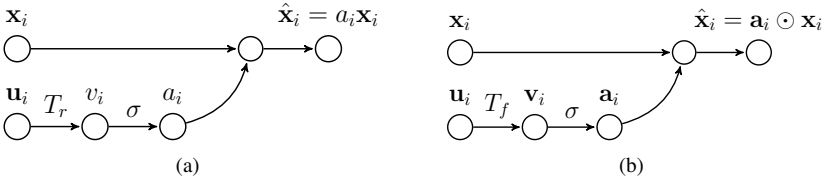


Figure 2: Illustration of (a) region-level gate unit and (b) feature-level gate unit.

region-level gate. In this figure, $\mathbf{x}_i$ denotes features from shared CNN networks for the i-th region of input image. In practice, it could be output from certain layer of the pre-trained CNN model. And $\mathbf{u}_i$ is an utility input used to control how important $\mathbf{x}_i$ is. It could be any data in the data-flow to generate $\mathbf{x}_i$. For instance, we find in our experiment that it works best if we respectively take the outputs of the 7-th and the softmaxed 8-th FC layers as $\mathbf{u}_i$ and $\mathbf{x}_i$ based on VGG networks structure [21]. The vector $\mathbf{u}_i$ is mapped to a probabilistic value a_i with feature space transformation $T_r : \mathcal{R}^{|\mathcal{U}|} \mapsto \mathcal{R}$ followed by a sigmoid function. Formally, the region-level gate operation is defined as

$$v_i = T_r(\mathbf{u}_i), \;\; a_i = \sigma(v_i), \;\; \hat{\mathbf{x}}_i = a_i\mathbf{x}_i \tag{2}$$

where $\hat{\mathbf{x}}_i$ is the feature values after gating operation. The mapping function T_r is realized by the fully connected layer. The sigmoid function yields a soft gate. As a result, contextual regions will be assigned with high gate values (close to 1) and keep their impacts for final classification, while irrelevant/noise regions will be suppressed. As gate units are integrated with deep neural networks, the parameters could be trained with the back-propagation (BP) algorithm. Gradients of the region-level gate are calculated by

$$\partial\mathbf{x}_i = a_i\partial\hat{\mathbf{x}}_i, \;\; \partial a_i = \mathbf{1}^T\partial\hat{\mathbf{x}}_i, \;\; \partial v_i = v_i(1-v_i)\partial a_i, \;\; \partial\mathbf{u}_i = T_r'(\mathbf{u}_i)\partial v_i \tag{3}$$

Feature-level gate controls the fine-grained pass/suppress for each feature dimension across different regions by their contribution to the classification. Figure 2(b) illustrates the structure of feature-level gate. In specific, the feature scaling is performed element-wise, instead of region-level. Considering that the dimensions of feature vector $\mathbf{x}_i$ and the gate control input $\mathbf{u}_i$ might not be the same, we first use a transformation function $T_f : \mathcal{R}^{|\mathcal{U}|} \mapsto \mathcal{R}^{|\mathcal{X}|}$ to map $\mathbf{u}_i$ to a new representation $\mathbf{v}_i$ with the same dimension of $\mathbf{x}_i$. Then the sigmoid function is applied to further convert them into probabilistic values $\mathbf{a}_i$ ranging from 0 to 1. Formally, the feature-level gate operation is defined as

$$\mathbf{v}_i = T_f(\mathbf{u}_i), \;\; \mathbf{a}_i = \sigma(\mathbf{v}_i), \;\; \hat{\mathbf{x}}_i = \mathbf{a}_i \odot \mathbf{x}_i \tag{4}$$

where $\odot$ means element-wise multiplication. Gradients of this kind of gate are calculated by

$$\partial\mathbf{x}_i = \mathbf{a}_i \odot \partial\hat{\mathbf{x}}_i, \;\; \partial\mathbf{a}_i = \mathbf{x} \odot \partial\hat{\mathbf{x}}_i, \;\; \partial\mathbf{v}_i = \mathbf{v}_i \odot (1-\mathbf{v}_i) \odot \partial\mathbf{a}_i, \;\; \partial\mathbf{u}_i = T_f'(\mathbf{u}_i)\partial\mathbf{v}_i \tag{5}$$

3.2 Fused Contextual Representation

After gate units, regional representations are aggregated to a unified image representation. Since the gate units already pick out contextual regions or features, here we apply order-less max-pooling to gated outputs, which simply calculates element-wise maximum values

across different regional features. To enhance the performance, we also introduce the multi-scale cross regions pooling scheme to group region features by their belonging region sizes. We divide regions into 5 scales according to the ratio of their sizes proportional to the size of the whole image, which are (0,1/32], (1/32, 1/16], (1/16, 1/8], (1/8, 1/4], (1/4, 1]. Then we perform cross-region max-pooling on gated region outputs separately at each scales. Finally, we concatenate features from each scale together to obtain a unified contextual representation. Thus, suppose $x_{i,k}$ is the k-th feature in the i-th region, the fused feature at dimension k of scale s is calculated as $z_{s,k} = \max\{x_{i,k} \mid R_i \in \mathcal{R}_s\}$, where $\mathcal{R}_s$ means the s-th region group. Note that this pooling scheme has the advantage to be seamlessly embedded into the whole classification network structure due to its simplicity, which is meanwhile proved to be very effective in our experiments.

3.3 RGNN Learning for Multi-label Classification

We feed aggregated contextual features to FC layers for classification purpose. Both gate units and classification FC layers are integrated into one deep neural network, so that we can learn the optimal parameters in an end-to-end manner. Specifically, the error propagation for contextual feature is assigned by $\partial x_{i,k} = \partial z_{s,k}$ if $i = \arg\max_j x_{j,k}$ for all $R_j \in \mathcal{R}_s$. In other word, $\partial x_{i,k} = \partial z_{s,k}$ if $x_{i,k}$ is the winner of maximum operation at k-th dimension across the s-th region group. Otherwise, $\partial x_{i,k}$ is set to 0.

We train the neural network using multi-label loss, with the last soft-max layer generating the estimated probabilities on target classes. In more details, cross entropy loss are calculated based on the differences between predicted values and ground truths, and the errors are back-propagated with mini-batch SGD method. Denote $\mathbf{y}$ as image label in the form of indicator vector of length C, where C is total number of classes. Here $y_j = 1$ if class j assigned to the image, otherwise $y_j = 0$. To deal with imbalanced datasets which are very popular in real world, we define a balanced loss function as follows

$$L_{cls} = \frac{1}{N}\sum_{i=1}^{N}\{\frac{1}{|\mathbf{y}_i = 1|}\sum_{j=1}^{C} y_{ij}\log\hat{y}_{ij} + \frac{1}{|\mathbf{y}_i = 0|}\sum_{j=1}^{C}(1-y_{ij})\log(1-\hat{y}_{ij})\} \quad (6)$$

where N is the mini-batch size and $\hat{y}_{ij}$ is the probabilistic output of RGNN. $|\mathbf{y}_i = 1|$ and $|\mathbf{y}_i = 0|$ count the number of ones and zeros in $\mathbf{y}_i$ respectively.

It is not necessary to train the RGNN network from scratch. For some tasks with relatively small datasets, we can start the network with the CNN models pre-trained on ImageNet. We can initialize the parameters for convolution layers and FC layers before cross-region pooling from the pre-trained CNN model, and other layer parameters with Xavier initialization [7], and then fine-tune the network on given training set.

When the object-level bounding-box annotations are available, we can introduce an additional task with localization loss. Our goal for localization in this paper is to pick out best object regions from region proposal candidates, rather than accurate bounding box regression as in [8]. We define the object localization loss as

$$L_{loc} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{T_i}\frac{1}{T_i}\|\mathbf{r}_{ij} - \mathbf{g}_{ij}\|^2 \quad (7)$$

where N is the mini-batch size and T_i is the number of regions in the i-th image. Here $\mathbf{g}_{ij}$ is the gate values on the j-th region and $\mathbf{r}_{ij}$ is the assumed ground truth importance of the corresponding region. The value of $\mathbf{r}_{ij}$ is given by the maximum IoU overlap ratio between

region R_{ij} and any ground truth regions $\mathcal{G}_i = \{G_{ik} \mid k = 1, \ldots, K\}$ in the image with a specified threshold θ. Mathematically,

$$\mathbf{r}_{ij} = \begin{cases} \mathbf{1}, & \text{if } \max\{\text{IoU}(R_{ij}, G_{ik}) | G_{ik} \in \mathcal{G}_i\} > \theta \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{8}$$

We simply set θ to 0.5 in our experiments. Note that $\mathbf{r}_{ij}$ and $\mathbf{g}_{ij}$ are vectors for feature-level gates, while they are scalars for region-level gates.

The two losses L_{cls} and L_{loc} are combined to a multi-task loss to train the network jointly for multi-label image classification and object localization:

$$L(x) = L_{cls}(x) + \lambda L_{loc}(x) \tag{9}$$

where λ is a hyper parameter controlling the weight of the combined localization losses.

4 Experiments

4.1 Datasets and Experimental Settings

We first evaluate the proposed method on the widely used PASCAL VOC 2007 and 2012 benchmarks for multi-label image classification. Both benchmarks contain images from 20 categories including animals, handmade objects and natural objects at wild locations and scales. In VOC 2007/2012 there are 5011/11540 images in trainval (short for training + validation) set and 4952/10991 images in test set. All images in the datasets are annotated with multiple class labels and object-level bounding boxes.

Then we make evaluation on the large-scale MS-COCO benchmark [14] in Section 4.5 to further verify the generalization capability of the proposed RGNN. MS-COCO dataset has a higher positive label density on a much larger label set of 80 common objects. It contains 82,783 images in the training set and 40,504 images in the validation set.

4.2 Implementation Details

The first few layers of RGNN were inherited from VGG-16 network pre-trained on ImageNet. We replaced the last max pooling layer in VGG-16 with the ROI pooling layer as introduced in [5], which projected differently sized ROI feature maps into fixed size ones. Then the fixed size feature maps were further connected to some FC layers. We adopted output of FC7 as gate input data for both region-level gate and feature-level gate, and imposed gate units to control the semantic outputs of FC8 after softmax transform. Both gate units consisted of one FC layer to map gate inputs to gate control values. Gated regional features were pooled into five scales as described in Section 3.2. In the end, two FC layers were applied to the fused contextual representation to produce classification results. Weights of all these new layers (gate units and classification FC layers) were initialized with the Xavier method [7].

We adopted a 4-step parameter-tuning scheme to help the network get properly trained. *First*, we fine-tuned VGG-16 network with size normalized input (without regions and ROI pooling) on 20-category VOC data using multi-label loss. This would give reasonable parameters initialization for RGNN. *Second*, we took free-sized images with regions as input and fine-tuned RGNN with ROI pooling, but with all gates turned on. *Third*, we trained the

Model Settings	mAP (%)
(1) VGG without fine-tune	89.3
(2) VGG + fine-tune	90.1
(3) Single-scale Regional VGG	89.9
(4) Multi-scale Regional VGG	90.8
(5) RGNN-RL + multi-label loss	92.9
(6) RGNN-RL + multi-task loss	93.7
(7) RGNN-FL + multi-label loss	93.1
(8) RGNN-FL + multi-task loss	93.7

Table 1: Ablation studies on VOC 2007.

Method	Property
PRE-1000C [7]	AlexNet + sliding windows.
CNN S TUNE [1]	Fine-tuned global CNN-S.
VGG-16+19 [21]	Fusion VGG16 + VGG19.
FV+LV-20-VD [20]	Fisher vector of regional CNN.
FV+LV-Fusion [20]	Fusion with VGG results.
DA-Fusion [15]	Fusion of deep attributes.
HCP-VGG [24]	HCP with VGG16.
HCP++ [24]	HCP-VGG + subcat. model.
SPD [16]	GoogleNet + part based fusion.

Table 2: List of compared methods.

network only with localization loss, which ensured a good initialization of gate units for next step. *Forth*, we fine-tuned the network with the proposed multi-task loss. In this step, we slowly decreased hyper parameter λ in Eq (9) from 1 to 0, allowing more freedom for the multi-label loss to select classification related contextual regions with reliable gate parameters. We fine-tuned the network for 10 epochs in the first 3 steps, and 20 epochs in the last step. Adam solver was used with the base learning rate set to 10E-4.

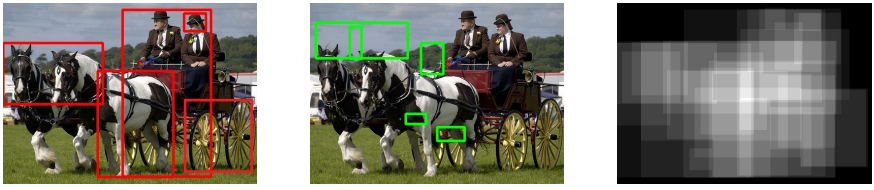
The system was implemented based on the Caffe framework [12], and trained on NVIDIA TitanX GPU. During testing, RGNN runs about *430ms* on GPU for each input image, including region proposal generating time with EdgeBox (∼200ms per image).

4.3 Ablation Studies

We explicitly investigate the contribution of different settings of the proposed methods in this subsection. First, we explore the proper network structure settings, especially on which layer is used for contextual feature pooling. On VOC 2007 test set, the mAP scores by contextual pooling over FC6/FC7/FC8/softmax-layer are 89.5/90.8/92.9/93.7 with region-level gates, and 89.0/92.3/92.9/93.7 with feature-level gates. This indicates that contextual pooling over deep softmax layer gives better results, which is consistent with the observations in [15].

Second, we gradually change the VGG-16 baseline structure towards RGNN structure, as listed in Table 1, and compare the performances of trained models in each step. More specific, Item-1 is the baseline VGG model without fine-tuning, but the result is obtained by evaluation on multiple image crops as in [21]. Item-2 is the VGG model globally fine-tuned on VOC 2007 training set. Item-3 is the VGG model with ROI pooling and single-scale cross region pooling, but with all gates tuned on. Item-4 is the VGG model with ROI pooling and multi-scale orderless cross region pooling, also with all gates tuned on. Item-5 is the proposed RGNN model with region-level gates, but fine-tuned only with multi-label loss. Item-6 is the RGNN model with region-level gates fine-tuned with the proposed multi-task loss. Items-7/8 are the feature-level gates counterparts of Item-5/6. It is obvious that RGNN achieves significant improvement over those models without gate units, and multi-task loss further improves the results in both cases. Comparing Item-3 and 4, it also shows that multi-scale contextual pooling used in our settings performs better than that of single-scale case.

Third, we visualize the outputs of region-level gates. Here we sort the gate values of all regions for the given image, and mark the top and bottom scored regions in Figure 3(a) and 3(b). We also build a heat-map of gate values on the whole image, as shown in Figure 3(c). It is obvious that the top regions (including person, horse, wheels, etc.) are more semantically meaningful and could be viewed as contextual regions, while bottom regions are trivial noisy regions. This study verifies the effectiveness of gate units for contextual region selection.



(a) (b) (c)

Figure 3: Illustration of (a) top-5 selected regions in red boxes; (b) bottom-5 discarded regions in green boxes; and (c) heat-map of region-level gate values mapped to the image.

4.4 Comparison with State of the Arts on VOC 2007 and 2012 Datasets

Here we compare the proposed RGNN method to most recent state-of-the-art methods on PASCAL VOC 2007 and 2012. We strictly follow the evaluation protocol by VOC and report performances by average precisions (AP). The results on VOC 2012 were obtained from online evaluation server. The compared methods and their property descriptions are listed in Table 2. All the compared methods utilize extra data during training or pre-training, such as ImageNet dataset and MS-COCO dataset. Except the first two methods and SPD on the table, all the others were based on the same deep VGG-16 model pre-trained on ImageNet. We elaborately add gate units based on VGG-16 in our RGNN experiments for fair comparisons. Note that VGG-16+19, FV+LV-Fusion, DA-Fusion and HCP++ are based on multiple models fusion from different cues, while our method is based on single model.

Table 3 and 4 respectively display the detailed per-category performances on VOC 2007 and 2012. It is obvious that RGNN with either region/feature-level gate achieves top results on both benchmarks. In specific, both RGNN-RL/FL achieve the best mAP of 93.7% on VOC 2007, while RGNN-RL achieves the best mAP of 93.4% on VOC 2012.

It's worth noting that RGNNs outperform the state-of-the-art single-model method HCP-VGG with a big margin of 2∼3% on mean-AP scores for both benchmarks. Remember that HCP-VGG also takes object region features into consideration. However, the region selection in HCP-VGG is based on heuristic rules. Compared to HCP-VGG, our RGNN takes advantages of automatically learnt gate units embedded in deep CNN classification networks to select important contextual regions. From the ablation study results reported in Table 1, we can see that all the introduced strategies like multi-scale cross regions pooling, gate units and multi-task learning contribute to the performance boost against HCP-VGG, while the proposed gate units bring the most significant performance gain.

It is shown in Table 4 that even compared to HCP++ (a multiple models fusion version of HCP) on VOC 2012, our proposed RGNNs are still slightly better in performance at a much faster processing speed compared to that of [24] (10s/image with VGG-16 on GPU). All these results demonstrate the effectiveness of RGNN for simultaneous contextual region feature selection and multi-label image classification.

4.5 Experiments on Large-Scale MS-COCO Dataset

In this part, we further evaluate RGNN on the much larger MS-COCO dataset [14] to examine its generalization capability. The only change to the RGNN structure compared to the settings in Section 4.4 was the expanded output layer dimension to 80 in order to agree with the new category size. As there are very few multi-label classification results yet reported on

																				mAP
PRE-1000C	88.5	81.5	87.9	82.0	47.5	75.5	90.1	87.2	61.6	75.7	67.3	85.5	83.5	80.0	95.6	60.8	76.8	58.0	90.4	77.9	77.7
CNN S TUNE	95.3	90.4	92.5	89.6	54.4	81.9	91.5	91.9	64.1	76.3	74.9	89.7	92.2	86.9	95.2	60.7	82.9	68.0	95.5	74.4	82.4
VGG-16+19	98.9	95.0	96.8	95.4	69.7	90.4	93.5	96.0	74.2	86.6	87.8	96.0	96.3	93.1	97.2	70.0	92.1	80.3	98.1	87.0	89.7
FV+LV-20-VD	97.9	97.0	96.6	94.6	73.6	93.9	96.5	95.5	73.7	90.3	82.8	95.4	97.7	95.9	98.6	77.6	88.7	78.0	98.3	89.0	90.6
FV+LV-Fusion	92.8	96.9	97.1	95.8	74.3	94.2	96.7	97.7	76.7	90.5	88.0	96.9	97.7	95.9	98.6	78.5	93.6	82.4	98.4	90.4	92.0
DA-Fusion	99.4	97.5	96.8	96.6	81.3	92.9	96.8	97.1	75.6	93.7	84.5	95.8	96.8	96.0	98.6	81.9	97.7	80.2	99.0	91.5	92.5
HCP-VGG	98.6	97.1	98.0	95.6	75.3	94.7	95.8	97.3	73.1	90.2	80.0	97.3	96.1	94.9	96.3	78.3	94.7	76.2	97.9	91.5	90.9
SPD	98.7	97.0	97.9	94.8	78.3	91.4	96.4	97.3	75.0	85.0	82.4	95.4	96.1	94.7	98.5	75.9	90.9	82.1	97.3	89.7	90.7
RGNN-RL	99.3	97.0	97.5	98.1	80.6	95.5	97.2	98.0	82.1	96.5	86.3	97.5	97.9	95.6	98.8	84.0	97.2	82.7	99.1	93.3	93.7
RGNN-FL	99.5	97.1	97.5	97.9	80.4	95.7	97.2	98.1	82.2	96.8	86.0	97.7	97.9	95.6	98.8	83.8	97.2	83.0	99.1	93.2	93.7

Table 3: Classification results (AP in %) comparison with state of the arts on VOC 2007.

																				mAP
PRE-1000C	93.5	78.4	87.7	80.9	57.3	85.0	81.6	89.4	66.9	73.8	62.0	89.5	83.2	87.6	95.8	61.4	79.0	54.3	88.0	78.3	78.7
CNN S TUNE	96.8	82.5	91.5	88.1	62.1	88.3	81.9	94.8	70.3	80.2	76.2	92.9	90.3	89.3	95.2	57.4	83.6	66.4	93.5	81.9	83.2
VGG-16+19	99.1	89.1	96.0	94.1	74.1	92.2	85.3	97.9	79.9	92.0	83.7	97.5	96.5	94.7	97.1	63.7	93.6	75.2	97.4	87.8	89.3
FV+LV-20-VD	98.4	92.8	93.4	90.7	74.9	93.2	90.2	96.1	78.2	89.8	80.6	95.7	96.1	95.3	97.5	73.1	91.2	75.4	97.0	88.2	89.4
FV+LV-Fusion	98.9	93.1	96.0	94.1	76.4	93.5	90.8	97.9	80.2	92.1	82.4	97.2	96.8	95.7	98.1	73.9	93.6	76.8	97.5	89.0	90.7
DA-Fusion	99.2	93.7	96.0	95.2	81.7	94.3	91.6	98.1	81.9	91.7	83.5	96.3	95.6	96.0	98.2	77.8	93.6	74.7	97.6	91.9	91.4
HCP-VGG	99.1	92.8	97.4	94.4	79.9	93.6	89.8	98.2	78.2	94.9	79.8	97.8	97.0	93.8	96.4	74.3	94.7	71.9	96.7	88.6	90.5
HCP++	99.8	94.8	97.7	95.4	81.3	96.0	94.5	98.9	88.5	94.1	86.0	98.1	98.3	97.3	97.3	76.1	93.9	84.2	98.2	92.7	93.2
RGNN-RL	99.3	95.7	97.7	95.4	84.5	96.2	94.6	98.4	84.6	95.6	84.1	97.9	98.0	96.7	98.7	82.9	96.1	79.6	98.6	93.4	93.4
RGNN-FL	99.3	95.4	97.6	95.2	84.7	96.1	94.6	98.4	84.7	95.4	84.0	97.9	98.0	96.8	98.8	82.6	95.9	79.3	98.5	93.7	93.3

Table 4: Classification results (AP in %) comparison with state of the arts on VOC 2012.

this new dataset, we simply followed the experimental settings in Section 4.3, aiming to test whether RGNNs could successfully improve results from global and region-level fine-tuned VGG networks (models without automatic region selection) on this dataset. We trained on the training set and made evaluation by mean-AP over all categories on the validation set. The global fine-tuned baseline VGG network had an mAP score of 65.8%. The region-level fine-tuned VGG network gained a better mAP of 69.7%. The proposed RGNN-RL and RGNN-FL further boosted the mAPs to 72.9% and 73.0% respectively. These results verified the effectiveness of RGNN on more complicated large-scale dataset.

5 Conclusion

We have proposed a novel deep regional gating neural network (RGNN) framework for multi-label image classification. Extensive experiments on popular benchmarks have clearly demonstrated that RGNN can boost classification performance due to its natural and effective information fusion from automatically selected contextual regions. Moreover, RGNN achieves this goal in an end-to-end learning manner with specially designed gate units and loss functions, thus making it a generic framework to work with various region proposal generation methods for improving image classification performance.

6 Acknowledgements

This work was supported in part by China's National 863 Program (#2014AA015101) and a grant from NSF China (#61572138). Part of this work was done when the first author was an intern at Intel Labs China.

References

[1] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *BMVC*, 2014.

[2] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. In *CVPR*, 2014.

[3] Mark Everingham, S M Ali Eslami, Luc Van Gool, Christopher K I Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *IJCV*, 111(1):98–136, 2014.

[4] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785, 2009.

[5] Ross B Girshick. Fast R-CNN. In *ICCV*, 2015.

[6] Ross B Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, pages 580–587, 2014.

[7] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[8] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale Orderless Pooling of Deep Convolutional Activation Features. In *ECCV*, pages 392–407. 2014.

[9] Chunhui Gu, Joseph J Lim, Pablo Arbeláez, and Jitendra Malik. Recognition using regions. In *CVPR*, pages 1030–1037, 2009.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[11] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. How good are detection proposals, really? *BMVC*, 2014.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, pages 1097–1105, 2012.

[14] Tsung-Yi Lin, Michael Maire, Serge J Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, pages 740–755, 2014.

[15] Jianwei Luo, Jianguo Li, Jun Wang, Zhiguo Jiang, and Yurong Chen. Deep Attributes from Context-Aware Regional Neural Codes. *arXiv.org*, 2015.

[16] Pascal Mettes, Jan C van Gemert, and Cees G M Snoek. No Spare Parts: Sharing Part Detectors for Image Categorization. *arXiv.org*, October 2015.

[17] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In *CVPR*, pages 1717–1724, 2014.

[18] Colin Raffel and Daniel P W Ellis. Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems. *arXiv.org*, 2015.

[19] Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, pages 91–99, 2015.

[20] Olga Russakovsky, Jia Deng, Hao Su, and et al. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[21] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv.org*, September 2014.

[22] Arnold W M Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE TPAMI*, 22(12):1349–1380, 2000.

[23] Jasper R R Uijlings, Koen E A van de Sande, Theo Gevers, and Arnold W M Smeulders. Selective Search for Object Recognition. *IJCV*, 104(2):154–171, 2013.

[24] Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. HCP: A Flexible CNN Framework for Multi-label Image Classification. *IEEE TPAMI*, pages 1–8, 2015.

[25] Ruobing Wu, Baoyuan Wang, Wenping Wang, and Yizhou Yu. Harvesting Discriminative Meta Objects with Deep CNN Features for Scene Classification. In *ICCV*, pages 1287–1295. IEEE, December 2015.

[26] Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, volume 37, 2015.

[27] Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. Can Partial Strong Labels Boost Multi-label Object Recognition? *arXiv.org*, 2015.

[28] C Lawrence Zitnick and Piotr Dollár. Edge Boxes: Locating Object Proposals from Edges. In *ECCV*, pages 391–405, 2014.