Information Entropy Based Feature Pooling for Convolutional Neural Networks

Weitao Wan, Jiansheng Chen*, Tianpeng Li, Yiqing Huang, Jingqi Tian, Cheng Yu, Youze Xue
Department of Electronic Engineering, Tsinghua University
wwt16@mails.tsinghua.edu.cn, jschenthu@mail.tsinghua.edu.cn
{ltp16, huang-yq17, tianjq16, yuc18, xueyz19}@mails.tsinghua.edu.cn

Abstract

In convolutional neural networks (CNNs), we propose to estimate the importance of a feature vector at a spatial location in the feature maps by the network's uncertainty on its class prediction, which can be quantified using the information entropy. Based on this idea, we propose the entropy-based feature weighting method for semantics-aware feature pooling which can be readily integrated into various CNN architectures for both training and inference. We demonstrate that such a location-adaptive feature weighting mechanism helps the network to concentrate on semantically important image regions, leading to improvements in the large-scale classification and weakly-supervised semantic segmentation tasks. Furthermore, the generated feature weights can be utilized in visual tasks such as weakly-supervised object localization. We conduct extensive experiments on different datasets and CNN architectures, outperforming recently proposed pooling methods and attention mechanisms in ImageNet classification as well as achieving state-of-the-arts in weakly-supervised semantic segmentation on PASCAL VOC 2012 dataset.

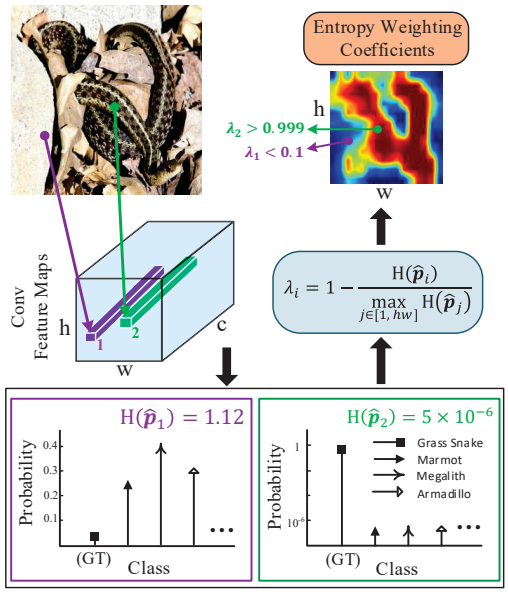

Figure 1. Illustration of the proposed location-adaptive feature weighting mechanism using information entropy. The model is trained with image classification labels. It's much more certain on its localized class predictions regarding the feature vector centered on the 'grass snake' than that on the background rock.

1. Introduction

Visual classification on large-scale image datasets [18, 38] is challenging because of the subtle inter-class differences and large background variations in natural images. The deep Convolutional Neural Networks (CNNs) is by far the most successful model for solving this problem and various CNN architectures [10, 31, 12, 30] have been proposed to continually boost the classification performances.

Current CNN models extract deep features from images by utilizing stacks of convolutional layers intersected with pooling layers for feature selection. To aggregate global semantics in the input image, in popular CNN architectures such as ResNet [10] and InceptionV3 [31], a Global Average Pooling (GAP) layer is placed between the last convolutional feature maps and the classifier. As such, these CNNs are capable of extracting global features which are robust to translation and rotations at very low computational costs.

In the GAP layer, the final image representation is computed as the average of all the feature vectors from the last convolutional feature maps. In other words, local features focused on different spatial locations in the input image are treated equally. Nevertheless, this may not be the best feature selection strategy. From the data distribution perspective, objects in natural images often appear on various complex backgrounds and there are frequent co-occurrences of different objects in the same image. At the same time, for feature vectors at different locations in the feature maps, the corresponding centers of the effective receptive field are different, indicating that semantics embedded in these feature

*Corresponding author

vectors may vary a lot. Some of these feature vectors are closely related to the target object and can contribute to the class prediction, while others may be uncorrelated or even harmful to the classification result. Hence, it is reasonable to aggregate information from different spatial locations in an adaptive manner rather than by averaging all the feature vectors across the spatial dimension with equal weights as GAP does. As such, feature vectors which are beneficial to the classification task should be augmented while noisy features should be suppressed.

A recent research [37] has revealed that in the CNN-GAP architecture, the importance of different image regions regarding a target class can be quantified as the dot products between the feature vectors and the classifier's weights, which is called the Class Activation Map (CAM). CAM method works as post-hoc additive to pre-trained models through inference for better understanding of the classification results. However, it is more desirable to design an internal module which can influence the training process by driving the network towards selecting features in a more effective way. To achieve this, the most critical problem is how to formulate a universal criterion for evaluating the feature importance before the classification result is produced.

We argue that this can be achieved by measuring the certainty of the network's class prediction. More specifically, the classifier can directly operate on each local feature vector to generate a localized class probability distribution. We employ the uncertainty measurement in the information theory and calculate the *information entropy* of this distribution. Intuitively, the entropy value characterizes how uncertain the network is on the classification of the corresponding local feature vector. Small entropy values indicate that the corresponding local feature vectors are probably correlated to object regions in the image, while large entropy values are usually associated with local feature vectors affected by misleading image regions, such as backgrounds or areas with confusing patterns, which are less helpful or even harmful to the final classification. Therefore, the importance of a local feature vector can be measured, for example, by a weighting coefficient negatively correlated to the corresponding entropy. Such an idea is illustrated in Fig. 1. We refer to this method as *Entropy Pooling* (EP), which can be readily integrated into popular CNN architectures.

We summarize our main contributions as follows:

- We propose a novel entropy-based mechanism to facilitate semantics-aware feature pooling, which helps the network to extract image representations that are more robust to background variations or interfering patterns.
- The EP's advantage of effectively locating semantically important regions and guiding the networks to concentrate on the most class-correlated regions is exploited to perform weakly-supervised localization and weakly-supervised semantic segmentation.
- Our approach is generic and can facilitate the performance for multiple vision tasks. By integrating the EP into various CNN backbones, our model outperforms recently proposed pooling methods and attention mechanisms in ImageNet [18] classification task. And we achieve new state-of-the-arts on the weakly-supervised semantic segmentation task on PASCAL VOC 2012 benchmark [8].

2. Related Work

Feature Pooling Recently proposed architectures, such as ResNet [10], InceptionV3 [31] and SENet [12] replace the computationally expensive FC layers in AlexNet [18] and VGG [30] with a GAP layer containing no trainable parameters. Several methods have been proposed to improve the simple average mechanism of GAP. In order to obtain the translation invariant and shape-preserving property, the 2D DFT-based pooling [26] computes the 2D DFT for each channel of the feature maps and then selects the magnitudes of the low frequencies as the new features. In the context of second-order pooling, the Bilinear Pooling [23] exploits the channel-wise correlations in the feature maps. However, on large-scale datasets, it is computationally expensive and has difficulty in robust covariance estimation given a small sample of very high-dimensional features. To address this issue, the FBP [21] leverages the factorized parameterization which introduces only a linear increase of parameters. The MPN-COV [20] is proposed to develop effective covariance-based pooling method on large-scale settings. And iSQRT-COV [19] improves the computational efficiency of MPN-COV by introducing an iterative matrix square root normalization method which is more suitable for parallel implementation on GPU.

Attention Mechanisms for Classification The proposed Entropy Pooling works by generating different weights for features at different locations. The attention mechanisms in CNNs follow similar practice. The Residual Attention Networks (RAN) [32] proposes to build the CNNs with the Attention Modules, which adopt a convolutional encoder-decoder architecture followed by sigmoid activation to generate attention maps to achieve spatial and channel-wise feature weighting. The Squeeze-and-Excitation Networks (SENet) [12] builds the SE module by averaging the intermediate convolutional feature maps along the spatial dimension, followed by two fully connected layers and sigmoid activation to generate channel-wise feature weighting coefficients, achieving state-of-the-art results on the ImageNet classification task. The main differences between our method and [32, 12] are that (1) we explicitly formulate the entropy weighting coefficients which are negatively correlated to the information entropy

of the localized class probability and (2) the entropy weighting coefficients can be reliably used to perform weakly-supervised localization and the networks are very effective in weakly-supervised semantic segmentation.

Weakly-supervised localization and semantic segmentation The Class Activation Maps (CAM) [37] can generate class scores distributed on the feature maps using only image-level labels during training. This mechanism is widely used to highlight the discriminative object parts for tasks such as the weakly-supervised localization and semantic segmentation. And the grad-CAM [27] is proposed to generalize its application in networks without GAP layer. Kolesnikov *et al*. [17] employs the CAM to locate the class-specific image regions and uses them as pseudo-labels for training the semantic segmentation networks. However, this supervision remains unchanged during the whole training process and they are small and sparse. To address this issue, the DSRG [13] is proposed to leverage seeded region growing for updating the initial segmentation masks (referred to as seed) during training. However, this method can be further improved by providing it with the initial seed of higher quality. The Multi-dilated Convolution (MDC) [35] employs multiple branches of convolutional layers with different dilation rates to transfer the class information from the most discriminative regions to the surroundings. However, as is shown in its experiments, it is difficult to transfer the class information on objects with a large area due to the size limit of the dilated convolution kernels. Our networks with the proposed Entropy Pooling can effectively locate the class-correlated areas and cover a large proportion of the true positive regions.

3. Entropy Pooling

In this section, we will explain the intuitions for developing the proposed approach. And we will present the formulation of the proposed Entropy Pooling method. Then we will describe how to insert it into intermediate convolutional layers with slight modifications.

3.1. Intuitions

We first revisit the prevailing CNN architecture composed of a cascade of convolutional layers, a GAP layer and a classifier which is typically a FC layer. The final prediction scores $\mathbf{F}$ over K classes before the softmax function are computed as Eq. 1, in which $f_{GAP}(\cdot)$ is the average operation of the GAP layer; $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K]$ is the weights of the classifier where each column corresponds to a class; $U \in \mathbb{R}^{h \times w \times c}$ is the last convolutional feature maps consisting of local feature vectors $\{\mathbf{v_i} \in \mathbb{R}^c | i = 1, 2, ..., hw\}$. For simplicity, we omit the classifier biases.

$$\begin{aligned} \mathbf{F} &= \mathbf{W}^T f_{GAP}(U) \\ &= \mathbf{W}^T \frac{1}{hw} \sum_i \mathbf{v}_i = \frac{1}{hw} \sum_i \mathbf{W}^T \mathbf{v}_i \end{aligned} \tag{1}$$

Each $\mathbf{v_i}$ actually encodes the semantic information mostly affected by a sub-region in the input image. Image sub-regions corresponding to different spatial locations in the feature maps may cover different object parts or background areas. This indicates that the correlations between different $\mathbf{v_i}$ and the target object class may vary significantly. However, under the CNN-GAP architecture, all the $\mathbf{v_i}$ are equally supervised by the class label due to the simple average operation of the GAP layer. We aim at exploring a spatially adaptive weighting scheme for the local feature vectors instead of simply averaging them.

3.2. Formulation

Our goal is to develop a semantics-aware pooling method by assigning different weights to different local feature vectors $\mathbf{v_i}$. We denote $\mathbf{W}^T\mathbf{v_i}$ in Eq. 1 as $\hat{\mathbf{F}}_i \in \mathbb{R}^K$, which is the classification score vector for location i in the feature maps. We call $\hat{\mathbf{F}}_i$ the localized class prediction of $\mathbf{v_i}$. Then we have $\mathbf{F} = \frac{1}{hw}\sum_i \hat{\mathbf{F}}_i$, indicating that performing classification on the GAP-pooled representation is equivalent to first obtaining the localized classification scores for each location in the feature maps and then averaging them. It is therefore possible to extract the semantic information encoded in each $\mathbf{v}_i$ before the pooling operation. Let $\hat{\mathbf{p}}_i = softmax(\hat{\mathbf{F}}_i)$ be the localized class probability for location i. The Shannon entropy [28] of $\hat{\mathbf{p}}_i$ can be calculated using Eq. 2.

$$H(\hat{\mathbf{p}}_i) = -\sum_{k=1}^{K} \hat{\mathbf{p}}_i(k) \log \hat{\mathbf{p}}_i(k) \tag{2}$$

For a feature location i, if its receptive field FOV_i is centered on a specific object, the localized class prediction of $\mathbf{v}_i$ made by the network should probably be highly confident, leading to a low $H(\hat{\mathbf{p}}_i)$ value. Otherwise, if FOV_i is centered on image textures or patterns frequently occurred in images of many different classes, the corresponding $H(\hat{\mathbf{p}}_i)$ should generally be high. A typical example is shown in Fig. 1. It's therefore reasonable to weight $\mathbf{v}_i$ using a coefficient λ_i negatively correlated to $H(\hat{\mathbf{p}}_i)$ as in Eq. 3.

$$\lambda_i = 1 - \frac{H(\hat{\mathbf{p}}_i)}{\max_{j \in [1,hw]} H(\hat{\mathbf{p}}_j)} \tag{3}$$

We refer to λ_i as the *entropy weighting coefficient* of feature location i; and the EP can be formulated as Eq. 4.

$$f_{EP}(U) = \frac{1}{hw} \sum_i \lambda_i \mathbf{v}_i \tag{4}$$

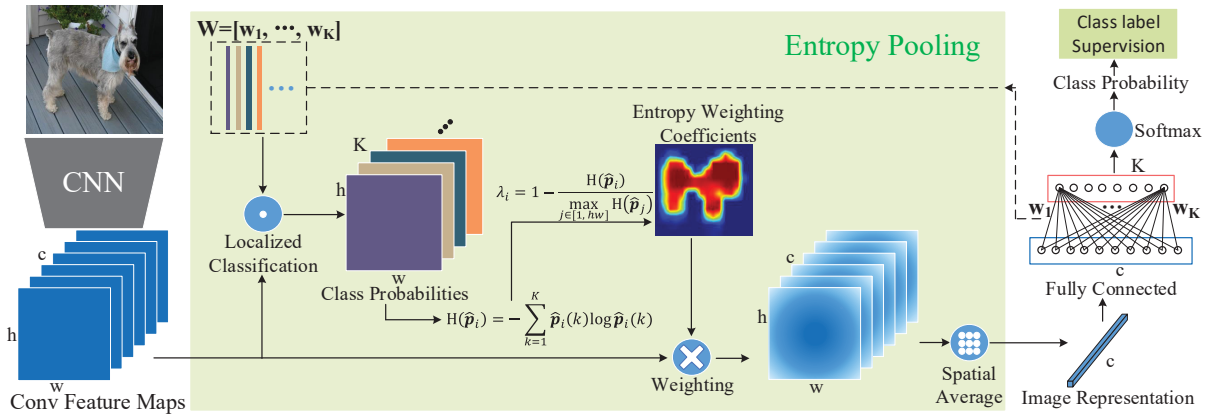

Figure 2. Architecture of the proposed EP layer which takes the last convolutional feature maps and classifier weights as input and perform localized classification. Then the entropy weighting coefficients λ_i are obtained using Eq. 3. The feature maps are weighted by multiplying λ_i and then averaged along the spatial dimension to produce the final representation.

The architecture of the EP layer is illustrated in Fig. 2. It should be noticed that the entropy based weighting process introduces no additional parameters. Therefore, EP can be used a post-hoc additive to pre-trained CNN models. For clarity, we refer to such a way of using EP as *EP inference*. Actually, we will show through experiments that even by using the EP inference, the classification performance can already be improved. A more favorable way of using EP is to incorporate it in training so as to guide the network to focus on making confident class predictions on more informative image regions by imposing stronger supervision on local features of greater importance.

3.3. Branched Entropy Weighting

According to the above formulation, EP layer can only be inserted between the last convolutional layers and the classifier. However, the core idea of EP, which is location-adaptive feature weighting, can actually be applied to intermediate convolutional layers to facilitate adaptive feature weighting in a hierarchical manner. Anyway, the EP layer need to be modified to adapt to the structure of the intermediate layers in the network.

According to Eq. 3, in order to obtain the entropy weighting coefficients, it's necessary to perform localized classification on the feature maps. However, the classifier weights $\mathbf{W}$ from the final FC layer cannot be directly applied to the feature maps of the intermediate layers. Therefore, a branched classification task has to be added. One possible solution is to directly fork a branched FC layer after the target intermediate layer to facilitate branched classification. In practice, the semantic information embedded in the intermediate feature maps may not be rich enough compared to the last feature maps, leading to the difficulty in training. To address this issue, the intermediate convolutional feature maps are further filtered by one convolutional layer to adapt for the classification. For efficiency, we follow the 'bottleneck' design [10] by utilizing a 1x1 convolutional layer with half the number of channels of the input features for dimension reduction. The FC layer can also be implemented using a 1x1 conv. layer to perform localized classification. Besides, we use no more than two BEW layers to achieve the best performance in the classification experiments. Overall, our EP does not introduce extra parameters and the BEW introduces only a small number of parameters.

We call such a modified version of EP the *Branched Entropy Weighting* (BEW) layer. The intermediate entropy weighting coefficients are calculated inside the BEW layer and then multiplied with the intermediate feature maps. A BEW layer and its connection to the backbone network is depicted in Fig. 3.

Multiple BEW layers can be integrated at different depths to the backbone network to form Multi-level BEW. As such, semantics-aware feature weighting are encouraged simultaneously at different levels of the network, leading to a further improvement in the final classification performance. Fig. 4 corresponds to a ResNet-50 network equipped with EP and two BEW layers. The classification losses of the backbone network and the two BEW branches are plotted. Also, the evolution of entropy weighting coefficients in the EP layer and the two BEW layers are presented. It can be observed that the generated entropy weighting coefficients gradually capture the semantically important image regions and covert the whole objects during training.

4. Experiments

To verify the effectiveness of our method, we conduct image classification experiments on two large-scale datasets, namely the ImageNet [18] and the Places365 [38], by integrating EP and BEW into various CNN architec-

tures. We perform the weakly-supervised localization task on ImageNet dataset by directly utilizing the generated entropy weighting coefficients. And we perform the weakly-supervised semantic segmentation task on PASCAL VOC 2012 dataset [8]. All the experiments are implemented using the Keras [6] API and Caffe [14] framework.

4.1. ImageNet Classification

The ImageNet ILSVRC 2012 dataset [18] contains 1.2M images for training and 50k images for validation, covering objects of 1000 classes. Following the practice of data augmentation in [10, 30, 31], each training image is resized with its shorter side randomly sampled in range $[342, 640]$ for the InceptionV3 and $[256, 480]$ for other architectures. Then we randomly crop a sub-region of size 299×299 for InceptionV3 and of size 224×224 for other architectures. The per-channel mean is subtracted and the images are horizontally flipped with a probability of 0.5 at each iteration during training. We initialize the network weights as in [9] and start training with a learning rate of 0.1, which is divided by 10 every 30 epochs. We train each model for 100 epochs using Stochastic Gradient Descent (SGD) with a momentum of 0.9, and a batch size of 256 on four Tesla P100 GPUs and the weight decay is set to 0.0001.

Network architectures We integrate the proposed EP and BEW to various CNN backbones, including VGG-16 [30], VGG-GAP [37], ResNet [10], InceptionV3 [31], among which the VGG-16 uses multiple FC layers to extract global semantics while the others utilize the GAP layer. For VGG-16, we keep the original layers intact by only adding BEW layers, without replacing the FC layers with the EP. There are 5 down-sampling operations with stride=2 in each of these CNN backbones. For VGG-16, we incorporate two BEW layers right before the 4th and the 5th pooling layers. For VGG-GAP, ResNet and InceptionV3, we replace the GAP layer with the EP layer after the last convolutional layer. In addition, we also incorporate two BEW layers in these three networks. The shallower one is placed right before the fourth down-sampling operation and the other is placed one convolutional layer (for VGG-GAP) or one residual/inception block (for ResNet/InceptionV3) before the GAP Layer. For baselines, we use the publicly released pre-trained models from the original papers. The

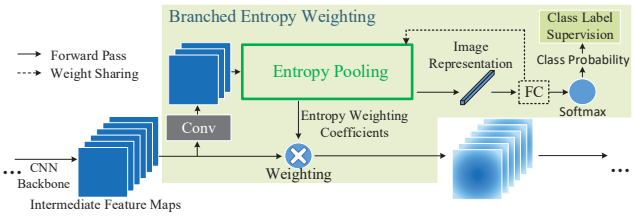

Figure 3. Architecture of the BEW layer which can be integrated into various CNN architectures at different intermediate layers.

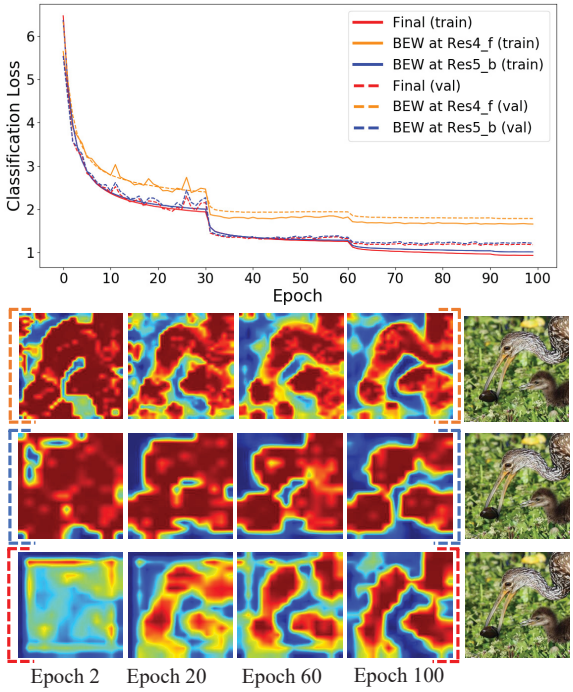

Figure 4. Illustration of the training progress for ResNet-50 integrated with the EP and BEW on the ImageNet dataset. The entropy weighting coefficients are visualized after being reshaped to image sizes for a sample validation image at 2, 20, 60, 100 epochs, respectively. The three rows from top to bottom correspond to two BEW layers and the EP layer from shallow to deep.

GAP layers in the baseline models are replaced with EP layers to perform *EP inference* experiments.

Evaluation We perform the standard 10-crop test [10, 18] on the validation set and report the top-1 and top-5 error rates. The evaluation results for different backbone networks are presented in Table 1. The classification performances of the original pre-trained models can be improved by simply utilizing the EP during inference. This verifies our intuition that EP is better than GAP by discriminating different local feature vectors. When the EP layer is adopted in training, the performances are further enhanced as expected. As we have pointed out, the EP layer guides the model to extract more robust representation by focusing on the semantically relevant regions and ignoring the noisy patterns. Multi-level BEW consistently achieves the best performances for all the four CNN architectures. Quite significant top-5/top-1 error reductions of $3.29\%/1.53\%$ are achieved for the VGG-16. The results demonstrate that our proposal generalizes well with various CNN architectures.

We compare our methods with state-of-the-art pooling and attention mechanisms in Table 2. For fair comparison, we perform the one-crop testing. The results demonstrate that the Branched Entropy Weighting and Entropy Pooling can effectively boost the performance of large-scale classi-

Configurations	VGG-16 [30]	VGG-GAP [37]	ResNet-50 [10]	InceptionV3 [31]
Baseline	28.07 / 9.33	31.66 / 11.28	22.85 / 6.70	20.20 / 5.13
EP inference (ours)	N/A	30.74 / 10.85	22.73 / 6.62	20.16 / 5.10
EP (ours)	N/A	30.37 / 10.62	22.64 / 6.52	20.07 / 5.06
Multi-level BEW + EP (ours)	**24.78 / 7.80**	**30.04 / 10.32**	**22.45 / 6.31**	**19.92 / 4.97**

Table 1. Top-5/Top-1 error rates (%, **10-crop** testing) on the ImageNet validation set. The 'N/A' indicates that we keep the original VGG-16 backbone intact without replacing its FC layers. The *Baseline* row refers to results reported from the original papers.

Method	Top-1 Err.	Top-5 Err.
Backbone: ResNet-50		
He *et al.* $_{\text{CVPR'16}}$ [10]	24.7	7.8
DFT-Pooling $_{\text{ECCV'18}}$ [26]	24.1	7.3
FBN $_{\text{ICCV'17}}$ [21]	24.0	7.1
SORT $_{\text{ICCV'17}}$ [33]	23.82	6.72
MPN-COV $_{\text{ICCV'17}}$ [20]	22.73	6.54
iSORT-COV $_{\text{CVPR'18}}$ [19]	22.14	6.22
iSORT-COV+EP (ours)	**21.97**	**6.02**
Backbone: ResNet-152		
He *et al.* $_{\text{CVPR'16}}$ [10]	23.0	6.7
Residual Attention $_{\text{CVPR'17}}$ [32]	21.76	5.9
SENet $_{\text{CVPR'18}}$ [12]	21.57	5.73
Backbone+BEW+EP (ours)	21.41	5.60
SENet+BEW+EP (ours)	**21.08**	**5.34**

Table 2. Top-5/Top-1 error rates (%, **one-crop** testing) on the ImageNet validation set, compared with state-of-the-arts of pooling methods and attention mechanisms for large-scale classification.

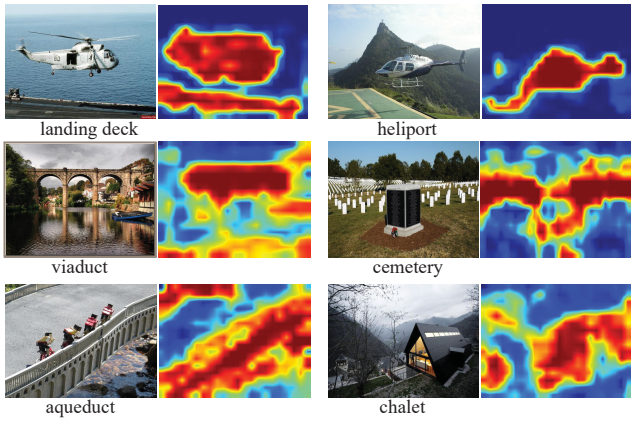

Figure 5. Visualization of the generated entropy weighting coefficients in the EP layer of ResNet-50 on Places365 validation set. All the samples are correctly classified by top-1 prediction. The first row shows two confusing classes. In the bottom right image, a small instance of chalet is also located at the bottom left corner.

fication and outperform other state-of-the-art methods using novel pooling methods or attention mechanisms in CNNs.

Fig. 4 helps to better understand how our proposal promotes the classification performances. The training and validation losses at different depths converge coherently during training. With the losses converging, the generated entropy weighting coefficients are becoming more and more promising and are gradually concentrating on the most important regions of the image. Both the quantitative and qualitative experiments verify that out proposal can boost the classification performances of different CNN models by driving them towards suppressing noisy patterns and emphasizing semantically important image regions.

4.2. Places365 Classification

We further evaluate our approach on another large-scale image classification benchmark, *i.e.* the Places365 dataset [38]. Places365 contains images from 365 different scenes, with over 1.8 million training images and 36,500 validation images. The input size is 224×224 for the CNNs and the data augmentation strategy is the same as [10]. Following the practice in [38], we use the CNNs pre-trained on ImageNet dataset for fine-tuning. The learning rate is initially set to 0.01 and divided by 10 every 30 epochs. We train each model for 100 epochs with a batch size of 256 on four GPUs. We use SGD with a momentum of 0.9 and the weight decay is set to 0.0001. We use the same network architectures described in Section 4.1.

Evaluation We perform 10-crop testing on the validation set and report the top-5 and top-1 error rates (%) in Table 3. It can be observed that the VGG-16+BEW model outperforms the original VGG-16 model by a margin of $1.40\%/1.45\%$. And the ResNet-50+BEW+EP model even over performs the ResNet-152 network which is significantly much deeper. Qualitatively, we also visualize the entropy weight coefficients for sample scene images in Fig. 5, which shows that the proposed method helps to locate the most semantically correlated features for scene recognition.

Network Architectures	Top-1	Top-5
AlexNet [38]	46.83	17.11
GoogLeNet [38]	46.37	16.12
ResNet-152 [38]	45.26	14.92
VGG-16 [38]	44.76	15.09
ResNeXt-101 [36]	43.79	13.75
ResNet-50 + BEW + EP (ours)	43.75	13.93
VGG-16 + BEW (ours)	**43.36**	**13.64**

Table 3. Top-5/Top-1 error rates (%, 10-crop testing) on the validation set of Places365.

Methods	top-1	top-5
Backprop on GoogLeNet [29]	61.31	50.55
VGGnet-GAP + CAM [37]	57.20	45.14
GoogLeNet-GAP + CAM [37]	56.40	43.00
ResNet-50 + CAM [37]	51.12	42.24
ResNet-50 + BEW + EP (ours)	**50.15**	**41.85**

Table 4. Weakly-supervised localization error rates (%) on the ImageNet validation set

4.3. Weakly-supervised Object Localization

As we can see, the entropy weighting coefficients generated by the EP layer outlines object locations quite accurately. Therefore, they can be utilized for the ImageNet weakly-supervised object localization task, in which only image-level labels are used for training. We employ the ResNet50+BEW+EP model trained on ImageNet in Section 4.1. Considering that the object localization task is class-dependent while the entropy weighting coefficients are not, we further define the class-specific entropy weighting coefficients in Eq. 5, in which c stands for the index of a specific class. λ_i can be interpreted as the objectiveness prior for feature location i. As such, λ_i^c can be regarded as the likelihood that location i belongs to object class c. We only consider the top 5 classes by summing up their λ^c's to get λ^{top5}. Because the possibility that objects outside of top 5 classes exist in the image is extremely low. Then we search the smallest rectangle which covers 98% of the total likelihood in λ^{top5} using the method proposed in [4] and use it as the predicted localization bounding box for the target classes.

$$\lambda_i^c = \lambda_i \hat{\mathbf{p}}_i(c),\ i \in [1, hw] \tag{5}$$

The generated λ and λ^{top5} for sample validation images are visualized in Fig. 6. It can be observed that λ^{top5} contains less noises than λ so that the object locations are more effectively captured. Numerical results for object localization are presented in Table 4. For fair comparison, we also apply the CAM method [37] to the ResNet-50 architecture which was not used in the original paper. Our method generally outperforms CAM, indicating that object regions are more accurately highlighted.

4.4. Weakly-supervised Semantic Segmentation

The proposed Entropy Pooling can guide the CNNs to extract semantic information by concentrating on most related object regions. It is combined with the Deep Seeded Region Growing (DSRG) method [13] which takes the localization maps for different classes as seeds for supervision and train the semantic segmentation networks while simultaneously growing the seeds. The seeds for foreground categories are originally derived by using the class activation maps (CAM) [37] on a fully convolutional variant of VGG-16 [30] networks which is trained with image-level multi-label supervision. We employ the EP at the last convolutional feature maps to facilitate semantics-aware feature pooling and effective localization for class-specific feature regions. More accurate and denser initial seeds can be generated because of this improvement, which can be observed in Figure 7. Then the seeds are fed into the DSRG method for seed growing and networks training.

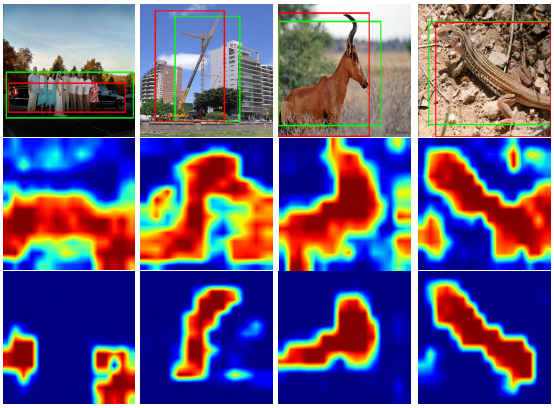

Figure 6. Weakly-supervised object localization with the predicted bounding boxes (green) and ground truth (red) presented. The second and third row visualize the original (λ) and the class-specific (λ^{top5}) entropy weighting coefficients respectively. In the top left image, the ground truth object is the car which is substantially occluded by a pile of people. Parts of the car that are not occluded are successfully highlighted in the corresponding λ^{top5} heatmap.

Dataset We evaluate the proposed approach on the PASCAL VOC 2012 segmentation benchmark dataset [8] which contains 20 object classes and a background class. Following common practice [5, 13], the training set is augmented to $10,582$ images. Only image-level class labels are used for all the experiments. We evaluate on both the validation and test set, which contains 1,449 and 1,456 images, respectively. The test set result is obtained by submitting the predictions to the official PASCAL VOC evaluation server.

Train/Test Settings The VGG-16 [30] networks pretrained on ImageNet [18] are used to initialize the multi-label classification networks and DeepLab-ASPP [5] semantic segmentation networks. We use SGD for training the classification and segmentation networks with momentum of 0.9 and weight decay of 0.0005. The batch size is 20 and the dropout ratio is 0.5. The learning rate starts from 0.001 and shrinks by a factor of 10 every 2000 iterations. Following DSRG [13], we use the saliency detection method [15] to locate the background pixels. Pixels belonging to top 30% of the largest class activation values in the heatmaps are considered as foreground object regions. And pixels whose normalized saliency values are smaller than 0.06 are considered as background. We use the public Caffe [14] implementation of DeepLab [5] and the networks are trained on a single NVIDIA GeForce GTX TITAN X GPU.

Method	Training	Val.	Test
Supervision: Box			
WSSL $_{\text{ICCV'15}}$ [25]	10K	60.6	62.2
BoxSup $_{\text{ICCV'15}}$ [7]	10K	62.0	64.2
GuidedSeg $_{\text{CVPR'17}}$ [24]	20k	55.7	56.7
Supervision: Spot			
1 Point $_{\text{ECCV'16}}$ [2]	10K	46.1	-
Scribblesup $_{\text{CVPR'16}}$ [22]	10K	51.6	-
Supervision: Image-level Labels			
SEC $_{\text{ECCV'16}}$ [17]	10K	50.7	51.7
STC $_{\text{TPAMI'17}}$ [34]	50K	49.8	51.2
TPL $_{\text{ICCV'17}}$ [16]	10K	53.1	53.8
DCSP $_{\text{BMVC'17}}$ [3]	10K	58.6	59.2
Hong *et al.* $_{\text{CVPR'17}}$ [11]	970K	58.1	58.7
AffinityNet $_{\text{CVPR'18}}$ [1]	10K	58.4	60.5
DSRG $_{\text{CVPR'18}}$ [13]	10K	59.0	60.4
MDC $_{\text{CVPR'18}}$ [35]	10K	60.4	60.8
DSRG+EP (Ours)	10K	**61.5**	**62.7**[1]

Table 5. Comparison of mIoU (%) for weakly supervised semantic segmentation methods on VOC 2012 validation and test sets.

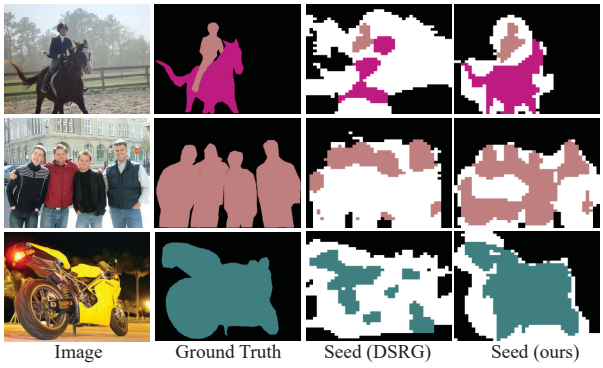

Figure 7. The seed generated by the plain networks used by DSRG [13] and networks with EP (ours). The category of the white pixels is either unknown or facing conflicts from the foreground heatmap and the background saliency result.

Comparisons with state-of-the-arts The results for weakly-supervised semantic segmentation solutions on PASCAL VOC validation and test sets are summarized in Table 5. It can be observed that our method outperforms previous state-of-the-art methods by a notable margin. Compared with the base networks DSRG, our method achieves a large improvement of 2.5% and 2.3% on the validation and test set. And we do not use extra training samples as the AffinityNet [1] and STC [34] do. Moreover, the EP does not introduce extra parameters for training, which is easy to implement and efficient in computation.

Qualitative Evaluation To demonstrate how the EP improves the performance when combined with DSRG, we visualize the seeds generated by the networks with and without the EP in Figure 7. Our model is effective in localizing the class-specific regions and produces high-quality seeds which maintain more shape information of the objects. While the networks without EP tend to capture only the most discriminative small discrete regions, which may be insufficient for training semantic segmentation networks. And we visualize some examples of the produced segmentation results of our model in Figure 8. It demonstrates that our method can produce satisfactory segmentation masks although only image-level labels are utilized for training.

[1] http://host.robots.ox.ac.uk:8080/anonymous/GRNBRX.html

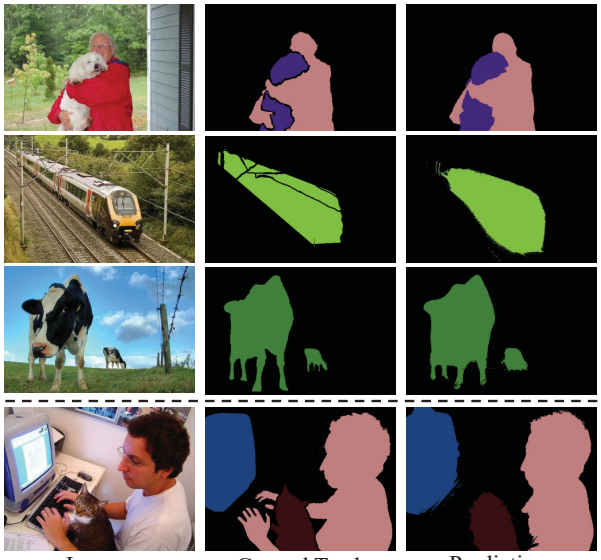

Figure 8. The segmentation results on the PASCAL VOC 2012 val set. The bottom row shows a failure case, which is caused by two factors. Firstly, hands are seldom highlighted for class 'person' in the seed generating step. Secondly, it's hard to grow the 'person' label into these hands because they are far from the upper body.

5. Conclusions

We propose a feature weighting mechanism based on information entropy to enable semantics-ware feature pooling in CNNs. Implemented as Entropy Pooing (EP) or the Branched Entropy Weighting (BEW) layer, our proposal enhances classification performances of different CNN models by guiding them to extract semantic information from more informative image regions without changing the backbone structures. Moreover, the networks with EP can generate high-quality seeds for weakly-supervised semantic segmentation and the Entropy Weighting Coefficients can be effectively employed for weakly-supervised localization. Extensive experiments on various datasets and CNN architectures verify the effectiveness of the proposed method.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (61673234).

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018.

[2] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. Whats the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[3] Arslan Chaudhry, Puneet K. Dokania, and Philip H. S. Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. *British Machine Vision Conference (BMVC)*, 2017.

[4] Jiansheng Chen, Gaocheng Bai, Shaoheng Liang, and Zhengqin Li. Automatic image cropping: A computational complexity study. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 507–515, 2016.

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

[6] François Chollet et al. Keras. `https://keras.io`, 2015.

[7] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015.

[8] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[9] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] Seunghoon Hong, Donghun Yeo, Suha Kwak, Honglak Lee, and Bohyung Han. Weakly supervised semantic segmentation using web-crawled videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7322–7330, 2017.

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[13] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018.

[14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[15] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2083–2090, 2013.

[16] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Two-phase learning for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3534–3543, 2017.

[17] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, pages 695–711. Springer, 2016.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Annual Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[19] Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 947–955, 2018.

[20] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2070–2078, 2017.

[21] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Factorized bilinear models for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2079–2087, 2017.

[22] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016.

[23] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[24] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. Exploiting saliency for object segmentation from image level labels. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5038–5047. IEEE, 2017.

[25] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015.

[26] Jongbin Ryu, Ming-Hsuan Yang, and Jongwoo Lim. Dft-based transformation invariant pooling layer for visual classification. In *Proceedings of the European Conference on Computer Vision*, pages 84–99, 2018.

[27] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra.

Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[28] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[29] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

[32] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

[33] Yan Wang, Lingxi Xie, Chenxi Liu, Siyuan Qiao, Ya Zhang, Wenjun Zhang, Qi Tian, and Alan Yuille. Sort: Second-order response transform for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1359–1368, 2017.

[34] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2314–2320, 2017.

[35] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7268–7277, 2018.

[36] Chen Yunpeng, Jin Xiaojie, Kang Bingyi, Feng Jiashi, and Yan Shuicheng. Sharing residual units through collective tensor factorization in deep neural networks. *arXiv preprint arXiv:1703.02180*, 2017.

[37] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *IEEE conference on Computer Vision and Pattern Recognition*, Jun 2016.

[38] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.