Neural Computing and Applications (2021) 33:7371–7380
https://doi.org/10.1007/s00521-020-05416-2

S.I. : DICTA 2019

Hierarchical class grouping with orthogonal constraint for class activation map generation

Fanman Meng[1] · Kaixu Huang[1] · Hongliang Li[1] · Shuai Chen[1] · Qingbo Wu[1] · King N. Ngan[1]

Received: 26 March 2020 / Accepted: 5 October 2020 / Published online: 3 November 2020

Abstract
Class activation map (CAM) generation aims at highlighting regions of a class in an image by the classification model. However, the regions obtained are usually small and local. Existing methods attribute the problem to the ineffective CAM extraction model and pay much attention on enlarging the regions via developing new models for CAM generation, but limited success has been made. Different from the existing methods, this paper attributes such incompleteness extraction to the finite discriminative cues within a single classification model and improves CAM generation by providing more discriminative cues via training multiple classification models with consideration of class relationships. To this end, the similarities between classes are firstly measured, and hierarchical clustering is then implemented to cluster initial clusters into multiple semantic meanings level of clusters. Afterward, multiple classification models are trained on these different levels of clustering, and multiple class activation maps with various and complementary discriminative cues are obtained. Finally, the class activation map is obtained via the combination of these maps. A new orthogonal module and a two-branch network for CAM generating are also proposed to improve CAM generation via making the regions orthogonal and complementary. Experimental results on the PASCAL VOC 2012 dataset show the superior performance of the proposed CAM generation method.

Keywords Class activation map (CAM) · Representative class selection · Orthogonal module

1 Introduction

Class activation map highlights the regions of a specific class in an image based on the classification model. It is an important task in computer vision as it can locate regions of a specific class from weak labels and thus can help many weakly supervised tasks such as segmentation [2, 7, 10, 12, 14, 25], detection [21, 24, 27, 29] and recognition [3, 5, 9, 20, 22].

The existing methods extract class activation map by two steps. The first one trains a classification network for all classes, and the second one highlights the regions based on the feedback of the classification network. However, these methods face the problem that the regions are usually small and local, such as the extraction of "Head" for object "Person" only, and many important regions are lost.

Solving such drawback has become an emerging research topic. Although researchers have proposed diverse CAM extraction methods [18, 30], they all attribute the problem to the ineffective CAM extraction method. Many efforts have been paid to enlarge the regions via developing new CAM generation method. For example, erase strategy [28] deletes regions already obtained and generates more regions by implementing the CAM generation again. Furthermore, some methods replace the classical convolution operator by a new convolution operator with a larger inception area, thus enlarging the regions. Although they improve the CAM generation reasonably and partially, the CAM generation, the experimental results reveal that the improvement is limited, i.e., CAM is still small and rough.

Different from the existing methods, we believe such a drawback is caused mainly by the single classification model, where discriminative cue cannot be provided sufficiently to activate more regions. When considering the

✉ Fanman Meng
fmmeng@uestc.edu.cn

[1] University of Electronic Science and Technology of China, Chengdu, China

classification of all classes in a single classification model, one class should be different from all the rest classes, which inevitably leads to the discriminative region small and local, just as the extraction results by the existing methods.

In order to prove such an assumption, some CAM results are displayed in Fig. 1, where class activation map of "Cat" based on four different settings of classification models is shown. The classification models are trained on PASCAL VOC dataset, and the setting "{Cat, Dog, Cow, Sheep, Horse}/5" means that the 20 classes are merged into five clusters, and "Cat" is within cluster "{Cat, Dog, Cow, Sheep, Horse}." Here, the cluster is formed by similar classes via clustering, and the classification model is trained based on the clustering results. One can see from Fig. 1 that the clusters containing "Cat" have different semantic levels in the four settings, and the classes for comparing with "Cat" vary. The class activation maps are shown from the second column to the fifth column, respectively. One can see the activation regions by the four models are different. Moreover, since these regions are complementary to each other, better results can be obtained by their combination, as shown in the sixth column. The results in Fig. 1 reveal that the CAM extraction is sensitive to the classes selected for comparing, and different regions can be obtained through varying the comparison classes. Through controlling the training classes by the class relationships, the desired and diverse regions can be obtained and better CAM extraction can be obtained via their combination.

Based on such motivation, this paper solves class activation map generation by the new view of forming multiple classification models to extract diverse and complementary class activation maps. In order to make the activated regions different and complementary, we form the classification models based on different levels of semantic meanings. We classify the initial classes into different semantic levels based on their similarity relationship using hierarchical clustering and extract the class activation maps based on different semantic levels so that the regions highlighted are different and complementary. Better results are obtained via the combination of these class activation maps.

Specifically, we propose a new CAM generation method consisting of four steps. In the first step, the similarities of classes are measured based on the classification network. In the second step, we cluster the initial classes into multiple levels of clusters via hierarchical clustering and train multiple classification models according to the clustering levels. In the third step, we extract CAMs from the classification models using a new two-branch classification network structure with the orthogonal constraint. Finally, we combine the class activation maps to form the final CAM. We verify our method on PASCAL dataset, and the

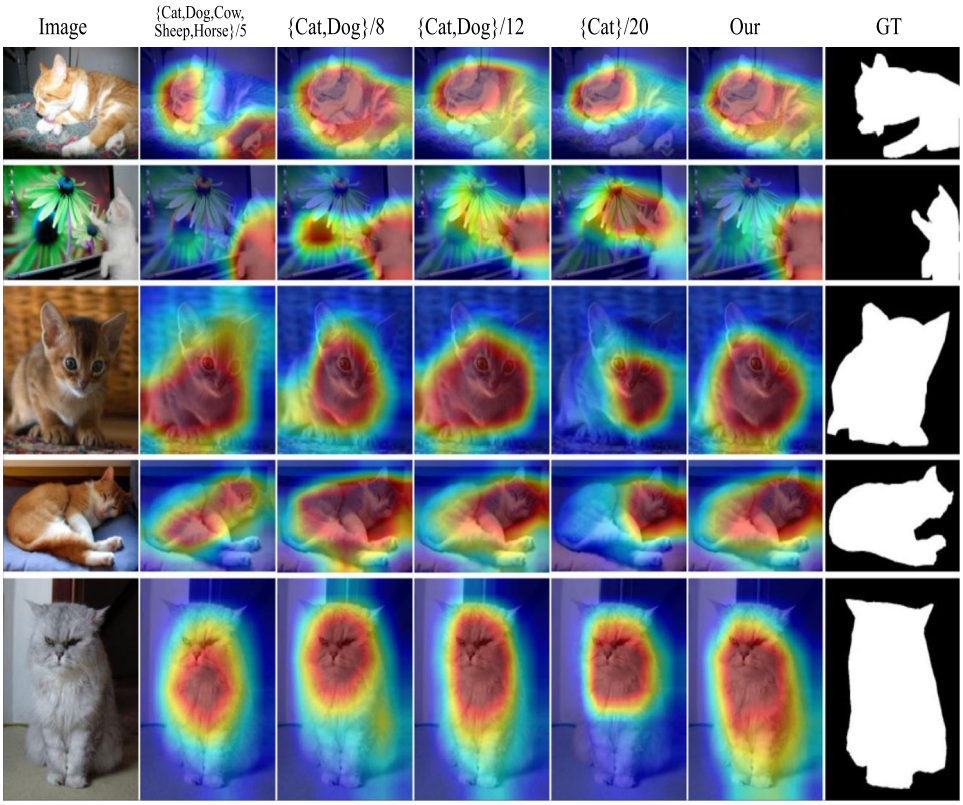

Fig. 1 First column: initial images. Second–fifth columns: the CAM results by different levels of clustering. Sixth column: the combination results. Last column: ground truth. $\{\cdot\}/N$: a cluster $\cdot$ by clustering classes into N classes

experimental results demonstrate the effectiveness of the proposed method.

The main contributions of the proposed method are listed as follows. Firstly, we implement class activation map by forming multiple classification models with different semantic meanings, which can capture more discriminative cues, and highlight different and complementary regions. Secondly, a hierarchical clustering method based on class-to-class relationships is proposed, and the clustering results of classes with different semantic meanings are obtained. Thirdly, a two-branch CAM generation method and the feature orthogonal constraint are proposed to obtain better CAM generation.

The rest of the paper is organized as follows. Section 2 illustrates the related work of the proposed method. The detailed steps are introduced in Sect. 3. Section 4 displays and discusses the experimental results. Finally, the conclusion is drawn in Sect. 5.

2 Related work

Class activation map generation is to extract regions of an image activated by a classification network for a specific class. On the one hand, it helps us understand the operations of the convolutional neural network in the process of classifying an image. On the other hand, since the classification network is trained by image-level labels only, it can be used in weakly supervised tasks to transfer image-level labels to pixel-level object regions. Therefore, it can help many weakly supervised tasks, such as image segmentation and detection.

The early CAM generation methods focus on generating class activation map via capturing the regions preserved in the top convolutional layer. In order to preserve spatial cues, some special layers such as global max pooling [16] and log-sum-exp pooling [17] are used to replace the FC layers that causes the loss of spatial information. In the method [30], average pooling layer with the final FC layer is used to replace the FC layers, and the weights between nodes of average pooling layers and nodes of FC layer (representing class label) are used to average the channel maps of the last convolution feature. Good localization is obtained. However, such a method changes the structure of the original network. To overcome such a drawback, the method in [18] proposes Grad-CAM which uses the gradients as the weights; therefore, the CAM can be extracted from the original classification networks directly.

However, the activated regions of those early methods are usually local and small, while many regions important to pixel-level tasks are lost. To enlarge the activation regions, several methods have been proposed. Erasing strategy which erases the regions already activated is an effective method to overcome such drawback. For example, after erasing the activated regions, the method in [23] uses the CAM extraction again to highlight activation regions in the rest regions. Similar to the method in [23], a two-phase-based CAM generation method is proposed in [11], which firstly obtains the activation regions in the first phase and then obtains the new regions from the rest regions in the second stage. The method in [13] searches activation regions based on the rule that the activation regions should mostly increase the classification error after erasing. The method in [28] searches complementary activation regions via adversarial complementary learning (ACoL). The method in [15] erases the activation regions to train a counterpart classifier with consideration of adversarial complementary attention. The method in [4] enlarges the activation map by attention-based dropout layer which hides the most discriminative part and informative region to extract more regions of objects. The method in [1] aims to expand the class activation map by random walk strategy with affinity matrix, which is learned by AffinityNet.

Although erasing strategy partially improves the CAM generation, the improvement is limited by the finite discriminative cues of the single classification model. It is the fact that the regions for distinguishing a class from all the rest classes are small and local. In this paper, we improve CAM generation by discovering more discriminative cues through forming new structure of classes by their similarity relationships.

3 The proposed method

3.1 Overview

Toward the goal of expanding the class activation map, we propose to mine more discriminative cues of a category by clustering original categories into multilevel hypercategories to generate better CAMs. The flowchart of the proposed method is shown in Fig. 2, including measuring the similarity of the classes, hierarchical clustering of the initial classes, generating CAM via a new two-branch framework and fusing the final CAM. In Sect. 3.2, the similarities of classes are calculated by the distances between their features, which is extracted from the classification network. In Sect. 3.3, we use a hierarchical clustering strategy to merge the most similar classes hierarchically, and cluster results with multiple semantic levels are obtained. In Sect. 3.4, we train the networks according to each level of clustering results via a new two-branch CAM generation framework with feature orthogonal module and generate multiple class activation maps for

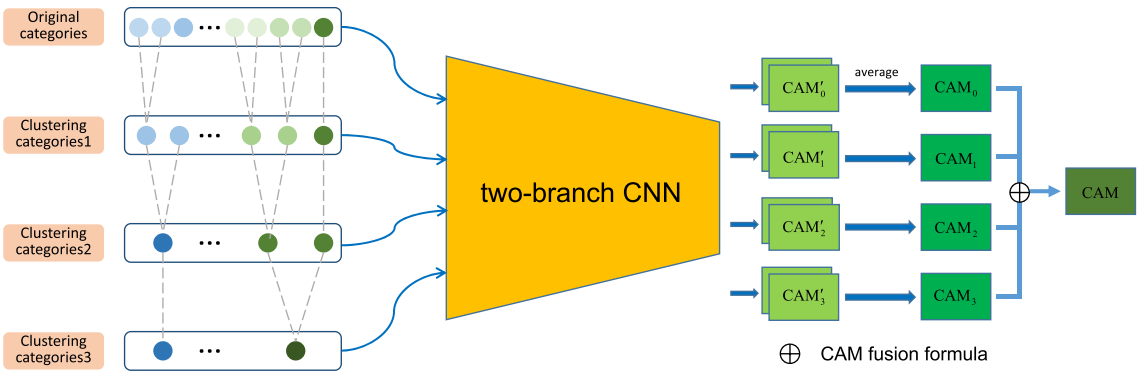

Fig. 2 Overview of our approach. It consists of four steps such as the class clustering, classification network training, CAM generation and fusion

each class of each image. In Sect. 3.5, we combine the CAMs to form the final CAM.

3.2 Measuring the similarities between classes

We calculate the similarity relationship between classes based on the distance of features, where the key is how to represent class by a feature that can describe classes well. To this end, we use deep classification network to represent the class with deep features. Specifically, we firstly train a classification network considering all classes. Then, by the fact that the weight vector of the node in the last FC layer, which maps the deep feature to the specific class labels, represents the combination manner of the class, we use it as the feature of class directly. Let n_c be the number of classes. We use the weight vector x_i of the ith node in the last FC layer as the feature of the ith class. By using n_k as the node number in the previous FC layer, each class is described by a $n_k \times 1$ vector x_i.

Given a pair of classes (c_i, c_j) with vectors (x_i, x_j), the Euclidean distance is used to calculate their similarity, i.e.,

$$d(x_i, x_j) = \|x_i - x_j\|_2 \tag{1}$$

3.3 Hierarchical clustering for initial classes

Since the single classification network used in the existing CAM generation methods contains finite discriminative cues, which limits the highlighting of more regions, we discover more discriminative cues through merging classes into groups under hierarchical semantic levels, so that the multiple classification models can provide discriminative cues that are different and complementary.

It is worth noting that the classes can be clustered randomly. Although such a method is simple, it ignores the important fact that the discriminative cues complementary to each other should be provided by the classes purposely selected. For example, the discriminative cues are more complementary for two pairs ("Cat," "Dog") and ("Cat," "Bus") than ("Cat," "Dog") and ("Cat," "Sheep"). By the fact that the classes have similarities in multiple semantic levels, we cluster the class hierarchically according to their semantic meanings.

Specifically, after representing each class c_i by the feature vector x_i, we perform the clustering by K-means algorithm with cluster number N, i.e., implementing the following two steps iteratively until convergence.

$$E\_step : a_j = \frac{1}{n_j^c} \cdot \sum_{i \in C_j} x_i \tag{2}$$

$$M\_step : \arg\min \left(\sum_{j=1}^{N} \sum_{i \in C_j} \|x_i - a_j\|_2 \right) \tag{3}$$

where a_j represents the cluster center of the jth cluster C_j and n_j^c represents the number of classes in the jth cluster.

After K-means clustering, we can obtain cluster set C^1. An example can be found in the bottom block of Fig. 3, where the initial 20 classes of PASCAL VOC dataset are clustered into 12 clusters. One can see similar classes, such as "Cat" and "Dog," are classified into one cluster. The clustering results mean that the similar classes are merged to form a new class with more rough semantic meanings, which can provide different discriminative cues compared with the initial class.

By using the same K-means-based clustering method, we further perform clustering on the clusters in C^1 to obtain new clustering results C^2 with more rough semantic meanings, as shown in the green block in Fig. 3, where the 12 clusters are clustered into eight clusters by the K-means algorithm. One can see the clusters "Sheep, Cow" and "Horse" are further merged, which indicates that classes with close semantic meanings can be clustered. We implement such clustering process further on the results of the second level to obtain the clustering results of the third level C^3, as shown in the purple block, where the 20 classes are finally clustered into five clusters.

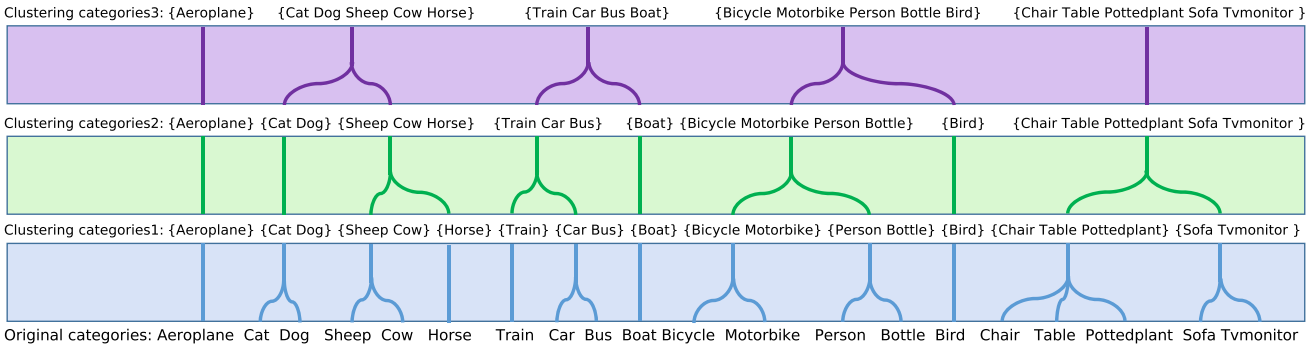

Fig. 3 Blue, green and purple represent the merging of categories for the first, second and third levels of clustering, respectively. All categories in curly brackets are considered as a cluster (color figure online)

Since one level of clustering is based on the clusters of the previous level, the results are hierarchically organized, as shown in Fig. 3. On the one hand, the semantic meanings of these levels are different, which can provide diverse discriminative cues. On the other hand, these cluster sets are organized in the hierarchical structure, which can guarantee the variations of the discriminative cues in a gradual manner.

3.4 Generating CAMs with feature orthogonal module

Based on the clustering results, we next train the classification model CNN^i for each level of clustering result C^i. The traditional methods mainly use single-branch-based network for the classification. Different from these methods, we use two-branch-based networks with different parameters to capture more discriminative cues. In addition, to enlarge the discriminative cues as much as possible, we force the two networks to generate different CAMs using the orthogonal constraint on the deep convolutional features.

3.4.1 Two-branch-based classification structure

The proposed two-branch-based classification network is shown in Fig. 4, where two branches CNN_1 and CNN_2 with different parameters are used to obtain features f_1 and f_2, followed by FC layer to obtain the classification results fc_1 and fc_2, respectively. The two outputs should be compared with the ground-truth labels, which are measured by classical classification losses L_{c1} and L_{c2}.

Since the two branches are forced to highlight different regions, feature orthogonal module is proposed to connect the two branches, which forces the descriptions of the two branches to be orthogonal, as shown in Fig. 4. Specifically, we propose feature orthogonal loss function to connect the two branches:

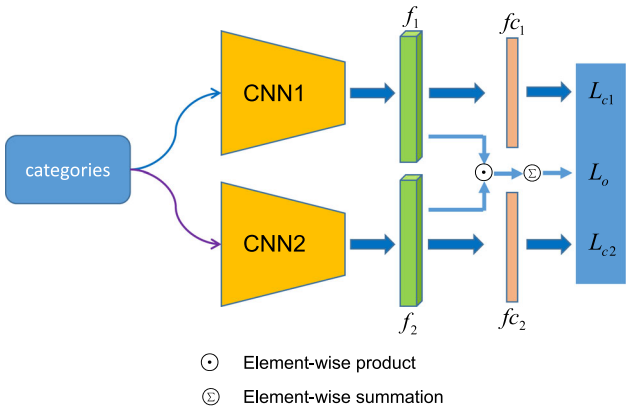

Fig. 4 Detailed illustration of our CAM extraction model. It consists of two subbranches that do not share parameters. We force the features of the two branches to be orthogonal by adding additional loss L_o

$$L_o = \|f_1 \odot f_2\|_{\mathrm{sum}} \tag{4}$$

where $\odot$ represents the Hadamard product between the features f_1 and f_2, and $\|x\|_{\mathrm{sum}}$ is the sum of all elements in x.

It is seen that the loss is large when the features are similar and is small otherwise. Hence, such loss penalizes the similar features and rewards dissimilar features.

So far, we define the overall loss function of the two-branch network as:

$$L_{\mathrm{all}} = L_{c1} + L_{c2} + \lambda L_o \tag{5}$$

where L_{c1} and L_{c2} represent the classical classification loss functions of the two branches, and λ is a parameter to balance the orthogonal loss and the classification loss.

Note that the orthogonal module has been used in [28] to highlight different class activation regions [28]. Our orthogonal module is different from the module in [28] due to the fact that the orthogonal module in [28] is formed in terms of the class activation map, i.e., making the extracted regions orthogonal. Meanwhile, we formulate orthogonal module by the feature space and make the deep feature

orthogonal, which can not only force the two networks to capture diverse cues, but also avoid failures caused by the same deep descriptions of the two networks.

3.4.2 CAM generation

Given an image and a classification model, the CAM is generated simply by two steps.

Step 1 Two CAMs M_1 and M_2 are extracted from the two branches using the CAM generation method [18].

Step 2 Two CAMs are combined to obtain the final CAM by the average operation, i.e., $M = \frac{1}{2}\sum_{i=1}^{2} M_i$.

3.4.3 Generating CAMs for multiple levels of clustering

Considering there are multiple levels of classification networks, the proposed two-branch CAM generation method is implemented on these classification networks to obtain multiple CAMs for each class of each image.

3.5 CAM fusion

Assuming the number of clustering level is k, a total of k number of classification networks are obtained. Therefore, a total of k number of CAMs are generated.

We next combine these CAMs to obtain the final CAM. For an image, the CAM of a class such as "Cat" is obtained by

$$M = M_0 + \frac{1}{k-1}\sum_{i=1}^{k-1} M_i - M_0' \tag{6}$$

where M_0 is the CAM of the classification model CNN_0 based on the initial classes, and M_i is the CAM by the ith-level clustering results for the class such as "Cat." M_0' is CAM of the rest classes.

4 Experiment

4.1 Experimental setup

4.1.1 Dataset

The PASCAL VOC 2012 [6] is used to verify our method, which consists of 20 object categories. Training dataset with 10,582 images and validation dataset with 1449 images are both employed.

4.1.2 Implementation details

In image normalization, the short side of the image is resized to 224. Then, the center area with size 224×224 is cut out from the image as the normalized image.

The classification network is initialized by the CNN models pre-trained on the ImageNet. We train our network on the NVIDIA GeForce GTX1080 with 8GB memory and PyTorch 0.4 framework. We set the initial learning rate to 0.0001. When the decrease in the loss within five epochs is smaller than a threshold, we reduce the learning rate by ratio 0.5. The batch size is set to 20. Each classification network is trained by 100 epochs.

In the step of obtaining the feature vector of class, we use ResNet-50 [8] as the backbone network to train the classification model. In order to balance the performance and training burden, we cluster the categories by four levels, and specifically, the cluster numbers for the four levels are set to 20, 12, 8 and 5, respectively.

For the loss function, we set L_{c1} and L_{c2} as binary cross-entropy loss and set the hyperparameter $\lambda = 0.0001$ for L_o.

4.1.3 Evaluation criteria

Two widely used metrics are employed to evaluate the performance of our approach and the comparison methods: mean intersection over union (mIoU) and mean localization error values (mLEV). MIoU is the standard measure of semantic segmentation, which measures the fitness between CAM and ground truth (larger is better), i.e.,

$$\mathrm{mIoU} = \frac{1}{n}\cdot\sum_i p_{ii} \Big/ \left(p_i + \sum_j p_{ji} - p_{ii}\right) \tag{7}$$

where n is the number of classes, p_{ji} is the number of pixels of class j predicted to belong to class i and p_i is the total number of pixels of class i.

The mLEV indicates the ratio of the inaccuracy in the localization of the CAM (lower is better), which is defined as

$$\mathrm{mLEV} = \frac{1}{n}\cdot\sum_k \min_i \min_m f(b_i, B_{km}) \tag{8}$$

where n is the number of all images and b_i is the algorithm generating bounding box. The ground-truth bounding boxes are $B_{km}, m = 1 \ldots M_k$, where k represents the kth class and M_k is the number of all instances of the kth class in the current image. If the overlap between b_i and B_{km} is more than 50%, $f(b_i, B_{km})$ will be 0, and 1 otherwise.

Since CAM is a probability map rather than a binary mask, the threshold $T = 0.15$ commonly used in CAM evaluation is also used to binarize the class activation map.

4.2 Subjective results

The class activation maps of our approach are shown in Fig. 5, where the original images, the CAMs of the multiple levels of clustering, the CAMs without clustering (i.e., the baseline method) and the CAMs by the proposed method are displayed. It can be seen that the activation maps of different clustering levels are different. In addition, the class activation map of the proposed method is obviously better than the CAMs of clustering levels and the baseline Grad-CAM [18], which proves the effectiveness of our approach on capturing more discriminative regions.

The CAM results of the two-branch-based network are displayed in Fig. 6, where (a) and (g) are input images and ground truth, respectively. (b) and (c) are the results of the two-branch network without the orthogonal module, which is equivalent to training two classification networks independently. (d) and (e) are the results of the two-branch network with the orthogonal module, and (f) is the results of the fusion of (d) and (e). It can be seen that the two-branch network with the orthogonal module is superior to the one without orthogonal module.

4.3 Objective results

4.3.1 The results by different clustering settings

We first display the results of the proposed method by different clustering settings. The results are shown in Table 1, where the number N in the first column presents the clustering setting. For example, $N = \{20, 12, 8\}$ means a three-level clustering by clustering the classes into clusters with cluster numbers 20, 12 and 8. The second column presents the mIoU values of the training set and the validation set, and the third column presents the mLEV values of the training set and the validation set. Quality measures are average for all images in training set or validation set of PASCAL VOC 2012.

As can be seen from Table 1, the mIoU values of the baseline (i.e., $N = \{20\}$) on the validation and training sets are 23.59% and 28.37%, and the mLEV values are 68.50% and 64.49%. When the clustering settings in our model are set to $\{20, 12\}$, $\{20, 12, 8\}$, $\{20, 12, 8, 5\}$, $\{20, 12, 8, 5, 2\}$, the mIoU values on the validation dataset are improved to 24.73%, 25.75%, 25.97% and 25.53%, and the mLEV values are improved to 67.86%, 65.77%, 65.08% and 65.81%, respectively. Moreover, mIoU values on the training set are improved to 29.15%, 29.67%, 30.24% and 29.68%, respectively, and the mLEV values are improved to 63.70%, 63.02%, 61.82% and 62.56%. It can be seen the CAM generation is improved by using clustering strategy.

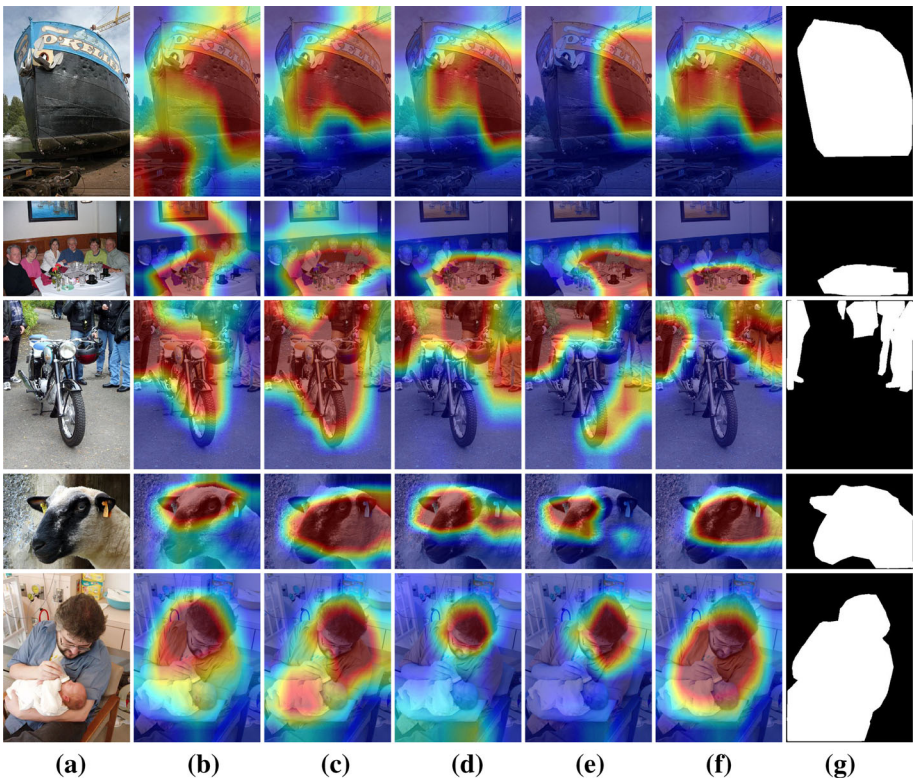

Fig. 5 **a** Input image; **b**–**e** results of the different clustering levels with cluster numbers 5, 8, 12, 20, respectively. **e** is the result of baseline; **f** results of our approach; **g** ground truth. By comparing the results from Column (**b**) to Column (**f**), one can easily observe that our approach using relationship between categories improves the results of CAM obviously

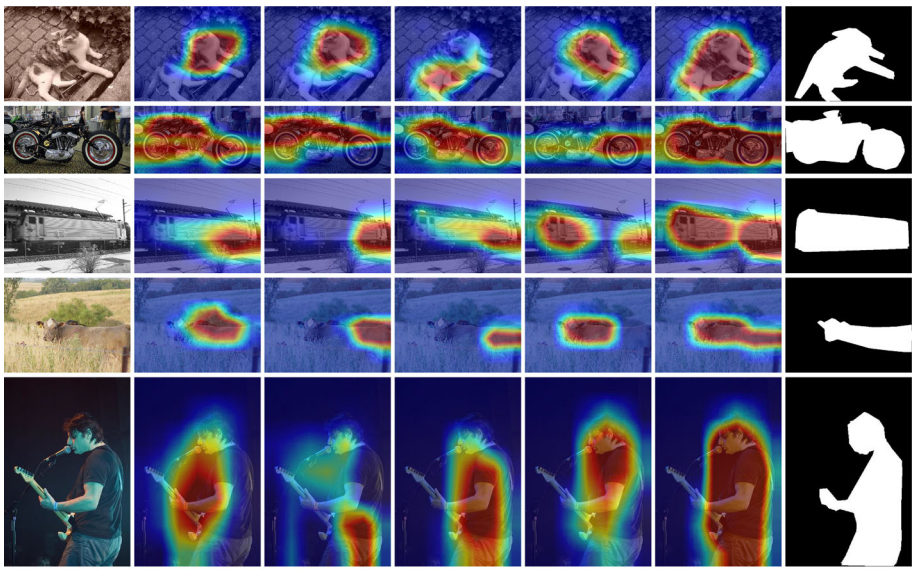

Fig. 6 **a** and **g**: input image and ground truth. **b** and **c**: the CAM results of the two-branch network without the orthogonal module. **d** and **e**: the results of the two-branch network with the orthogonal module. **f**: the CAM result by combining (**d**) and (**e**)

Table 1 mIoU values and mLEV values by the different clustering settings of our methods

Hierarchical clustering setting	mIoU	mLEV
Baseline $N = \{20\}$ [18]	28.37/23.59	64.49/68.50
$N = \{20, 12\}$	29.15/24.73	63.70/67.86
$N = \{20, 12, 8\}$	29.67/25.75	63.02/65.77
$N = \{20, 12, 8, 5\}$	**30.24/25.97**	**61.82/65.08**
$N = \{20, 12, 8, 5, 2\}$	29.68/25.53	62.56/65.81
$N = \{20, 15\}$	30.05/25.56	62.02/65.71
$N = \{20, 15, 11\}$	30.32/25.86	61.70/65.19
$N = \{20, 15, 11, 8\}$	**30.43/26.08**	**61.63/64.92**
$N = \{20, 15, 11, 8, 6\}$	30.14/25.91	61.91/65.17
$N = \{20, 10\}$	30.12/25.83	61.96/65.23
$N = \{20, 10, 5\}$	**30.24/25.92**	**61.78/65.15**
$N = \{20, 10, 5, 3\}$	29.57/25.53	62.59/65.50
$N = \{20, 10, 5, 3, 2\}$	28.20/24.25	64.67/67.13

Bold indicates the best setting of given clustering numbers

Meanwhile, the results of $N = \{20, 12, 8, 5\}$ are the best compared with $\{20, 12, 8, 5, 2\}$. This indicates that setting large number of levels is harmful to the CAM generation.

Additional experiments are conducted to show the impact of different clusters of the specific hypercategory level on the performance. We validate the performance of two another serial settings: one series is $\{20, 15\}$, $\{20, 15, 11\}$, $\{20, 15, 11, 8\}$, $\{20, 15, 11, 8, 6\}$, and the other series are $\{20, 10\}$, $\{20, 10, 5\}$, $\{20, 10, 5, 3\}$, $\{20, 10, 5, 3, 2\}$. The experiments show there is slight performance gain when the number of the clusters is set to $\{20, 15, 11, 8\}$. And while the number of clusters is set to $\{20, 10, 5, 3\}$, the performance drops from 25.97% to 25.53% and even drops to 24.25% when two additional clusters are included, resulting from that the new cluster number is too small to catch the discriminative cues. More details can be found in Table 1.

4.3.2 Comparisons with existing methods

We compare our approach with several existing CAM generation methods, such as CAM [30],[1] Grad-CAM [18], ACoL [28][2] and CBAM [26].[3] We use the code published by the author to train the model. For CAM and Grad-CAM, we use ResNet-50 and VGG-16 [19] as backbone networks to generate CAM. For ACoL, we use the VGG-16 recommended in the code as the backbone. For CBAM, ResNet-50 is used (for fair comparison) as the backbone network.

The mIoU and mLEV values of the existing methods and our method are shown in Tables 2 and 3. It is seen from the tables that the proposed method is superior to the existing methods on both training and validation datasets, because our method can capture more discriminative cues by using multiple semantic level of classification models, which results in the generation of more complementary and better class activation maps.

4.3.3 Ablation study

In this subsection, we conduct ablation study on our approach. Table 4 shows the results of the ablation study by

[1] https://github.com/metalbubble/CAM.

[2] https://github.com/xiaomengyc/ACoL.

[3] https://github.com/Jongchan/attention-module.

Table 2 mIoU values by the proposed method and the comparison methods

Method	Network	PASCAL-val	PASCAL-train
CAM [30]	ResNet-50	20.90	24.69
Grad-CAM [18]	ResNet-50	23.59	28.37
CBAM [26]	ResNet-50	19.93	23.85
Our	ResNet-50	**25.97**	**30.24**
CAM [30]	VGG-16	23.81	27.35
Grad-CAM [18]	VGG-16	27.62	30.79
ACoL [28]	VGG-16	19.52	20.41
Our	VGG-16	**31.32**	**34.37**

Bold indicates the best performance

Table 3 mLEV values by the proposed method and the comparison methods

Method	Network	PASCAL-val	PASCAL-train
CAM [30]	ResNet-50	74.33	70.32
Grad-CAM [18]	ResNet-50	68.50	64.49
CBAM [26]	ResNet-50	77.04	71.47
Our	ResNet-50	**65.08**	**61.82**
CAM [30]	VGG-16	68.40	65.57
Grad-CAM [18]	VGG-16	65.62	63.24
ACoL [28]	VGG-16	76.56	75.32
Our	VGG-16	**58.84**	**55.89**

Bold indicates the best performance

Table 4 mIoU values and mLEV values of the ablation studies of our methods

Clustering	Orthogonal	mIoU	mLEV
×	×	28.37/23.59	64.49/68.50
×	✓	29.34/24.32	62.84/67.26
✓	×	29.04/24.95	63.19/67.53
✓	✓	**30.24/25.97**	**61.82/65.08**

Bold indicates the best performance

whether using the category clustering method or the feature orthogonal module or not. The manners of their combinations are shown in the first and second columns. The third to fourth columns present the mIoU and mLEV values of the training set and the validation set.

The results of the ablation experiments of our proposed method are shown in Table 4. When using the proposed feature orthogonal module only, mIoU values on the validation and training sets increase by 0.73% and 0.97% to the baseline method (without using both the clustering and orthogonal methods), and mLEV values on the validation and training sets decrease by 1.24% and 1.65%, respectively. When using the category clustering method only, the mIoU and mLEV values are both improved (1.36% and 0.67%, and 0.97% and 1.30%, respectively). When using both the category clustering method and the feature orthogonal module, mIoU and mLEV values are further improved (2.38% and 1.87%, and 3.42% and 2.67%, respectively), which demonstrates the usefulness of the proposed method using clustering strategy and orthogonal module simultaneously.

5 Conclusion

This paper proposes a new class activation map generation method, which extracts CAM by multiple-level class grouping and orthogonal constraint. A hierarchical clustering method based on class relationships is firstly proposed to cluster classes into multiple levels of clusters, in order to capture diverse discriminative cues. Then, the clusters are treated as new classes to train multiple classification networks. To generate CAM more accurately, a new two-branch-based network is proposed for training, and an orthogonal module forcing feature orthogonal is proposed to obtain diverse CAMs of the two branches. Finally, the fusion method is proposed to combine the CAMs of the multiple networks and generate the final CAM. The experimental results show that our method improves CAM generation in terms of larger mIoU values and smaller mLEV values.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grants 61871087, 61502084, 61831005 and 61601102 and supported in part by Sichuan Science and Technology Program under Grant 2018JY0141.

References

1. Ahn J, Kwak S (2018) 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 (CVPR), pp 4981–4990
2. Araslanov N, Roth S (2020) Single-stage semantic segmentation from image labels. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4253–4262
3. Chen X, Xu C, Yang X, Tao D (2018) Attention-GAN for object transfiguration in wild images. In: Proceedings of the European conference on computer vision (ECCV)
4. Choe J, Shim H (2019) Attention-based dropout layer for weakly supervised object localization. In: The IEEE Conference on computer vision and pattern recognition (CVPR)
5. Dubost F, Adams H, Yilmaz P, Bortsova G, van Tulder G, Ikram MA, Niessen W, Vernooij MW, de Bruijne M (2020) Weakly

supervised object detection with 2d and 3d regression neural networks. Med Image Anal 65:101767
6. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The PASCAL visual object classes challenge 2012 (VOC2012) results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
7. Ge W, Yang S, Yu Y (2018) Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In: IEEE conference on computer vision and pattern recognition (CVPR) (2018)
8. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition, pp 770–778
9. He X, Peng Y, Zhao J (2018) Fast fine-grained image classification via weakly supervised discriminative localization. IEEE Trans Circuits Syst Video Technol 29(5):1394–1407
10. Jiwoon A, Cho S, Suha K (2019) Weakly supervised learning of instance segmentation with inter-pixel relations. In: IEEE conference on computer vision and pattern recognition (CVPR)
11. Kim D, Cho D, Yoo D (2017) Two-phase learning for weakly supervised object localization. In: The IEEE international conference on computer vision (ICCV), pp 3554–3563
12. Lee J, Eunji K, Lee S, Lee J, Sungroh Y (2019) FickleNet: weakly and semi-supervised semantic image segmentation using stochastic inference. In: IEEE conference on computer vision and pattern recognition (CVPR)
13. Li K, Wu Z, Peng KC, Ernst J, Fu Y (2018) Tell me where to look: guided attention inference network. arXiv preprint arXiv:1802.10171 (2018)
14. Li Q, Anurag A, Torr PH (2018) Weakly- and semi-supervised panoptic segmentation. In: Proceedings of the European conference on computer vision (ECCV)
15. Li X, Liu J, Wang M (2019) Weakly supervised fine-grained visual recognition via adversarial complementary attentions and hierarchical bilinear pooling. In: Gedeon T, Wong KW, Lee M (eds) Neural information processing. Springer, Cham, pp 74–85
16. Oquab M, Bottou L, Laptev I, Sivic J (2015) Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 685–694
17. Pinheiro PO, Collobert R (2015) From image-level to pixel-level labeling with convolutional networks. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1713–1721
18. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D, et al (2017) Grad-CAM: visual explanations from deep networks via gradient-based localization. In: The IEEE international conference on computer vision (ICCV), pp 618–626
19. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations (ICLR)
20. Singh KK, Lee YJ (2017) Hide-and-seek: forcing a network to be meticulous for weakly-supervised object and action localization. In: Proceedings of the IEEE international conference on computer vision (ICCV). IEEE, pp 3544–3553
21. Wan F, Wei P, Jiao J, Han Z, Ye Q (2018) Min-entropy latent model for weakly supervised object detection. In: IEEE conference on computer vision and pattern recognition (CVPR)
22. Wang C, Zheng H, Yu Z, Zheng Z, Gu Z, Zheng B (2018) Discriminative region proposal adversarial networks for high-quality image-to-image translation. In: Proceedings of the European conference on computer vision (ECCV)
23. Wei Y, Feng J, Liang X, Cheng MM, Zhao Y, Yan S (2017) Object region mining with adversarial erasing: a simple classification to semantic segmentation approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1568–1576
24. Wei Y, Shen Z, Cheng B, Shi H, Xiong J, Feng J, Huang T (2018) TS2C: tight box mining with surrounding segmentation context for weakly supervised object detection. In: Proceedings of the European conference on computer vision (ECCV)
25. Wei Y, Xiao H, Shi H, Jie Z, Feng J, Huang TS (2018) Revisiting dilated convolution: a simple approach for weakly- and semi-supervised semantic segmentation. In: IEEE Conference on computer vision and pattern recognition (CVPR)
26. Woo S, Park J, Lee JY, Kweon IS (2018) Cbam: convolutional block attention module. In: European conference on computer vision (ECCV)
27. Xiaopeng Z, Jiashi F, Hongkai X, Qi T (2018) Zigzag learning for weakly supervised object detection. In: IEEE conference on computer vision and pattern recognition (CVPR)
28. Zhang X, Wei Y, Feng J, Yang Y, Huang TS (2018) Adversarial complementary learning for weakly supervised object localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1325–1334
29. Zhang X, Wei Y, Kang G, Yang Y, Huang T (2018) Self-produced guidance for weakly-supervised object localization. In: Proceedings of the European conference on computer vision (ECCV)
30. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929