LayerCAM: Exploring Hierarchical Class Activation Maps for Localization

Peng-Tao Jiang*, Chang-Bin Zhang*, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei

***Abstract*—The class activation maps are generated from the final convolutional layer of CNN. They can highlight discriminative object regions for the class of interest. These discovered object regions have been widely used for weakly-supervised tasks. However, due to the small spatial resolution of the final convolutional layer, such class activation maps often locate coarse regions of the target objects, limiting the performance of weakly-supervised tasks that need pixel-accurate object locations. Thus, we aim to generate more fine-grained object localization information from the class activation maps to locate the target objects more accurately. In this paper, by rethinking the relationships between the feature maps and their corresponding gradients, we propose a simple yet effective method, called LayerCAM. It can produce reliable class activation maps for different layers of CNN. This property enables us to collect object localization information from coarse (rough spatial localization) to fine (precise fine-grained details) levels. We further integrate them into a high-quality class activation map, where the object-related pixels can be better highlighted. To evaluate the quality of the class activation maps produced by LayerCAM, we apply them to weakly-supervised object localization and semantic segmentation. Experiments demonstrate that the class activation maps generated by our method are more effective and reliable than those by the existing attention methods. The source code is available at our project page: https://mmcheng.net/layercam/.**

***Index Terms*—Weakly-supervised object localization, class activation maps.**

RECENTLY, a lot of attention methods [1], [2], [3] have been proposed to utilize the CNN-based image classifiers to generate class activation maps. These maps can locate the regions of the target objects, where the pixels with strong values in them are more likely to belong to the target objects. As image-level labels only indicate whether the target objects exist, they do not provide any object location information. Thus, the localization ability of the class activation maps can make up such an issue of image-level labels, which further facilitates the ill-posed weakly-supervised tasks, such as weakly-supervised semantic segmentation [4], [5], [6], [7] and weakly-supervised object localization [8], [9] under image-level supervision.

The concept of class activation maps is firstly proposed in CAM [1]. They generate class activation maps by utilizing a specific network structure that replaces the fully-connected layer of the image classifier with the global average pooling layer. Later, Grad-CAM [2] enhances the generalization ability of this technique, which enables generating class activation maps for any off-the-shelf CNN-based image classifier possible. Grad-CAM utilizes the average gradients of a feature map to represent its importance to the target category. Although these methods can effectively locate the target objects, a common issue among them is that they all rely on the final convolutional layer of CNN to generate class activation maps. Due to the low spatial resolution of the output from the final convolutional layer, the resulting class activation maps can only locate coarse object regions. As shown in Fig. 1, the class activation maps generated by Grad-CAM from $conv5\_3$ of VGG-16 [10] can only locate the general location of the horse. They cannot obtain fine details of the horse, such as the location of the horse leg. However, the weakly-supervised tasks, such as semantic segmentation, usually need more accurate object localization information. Class activation maps from the final convolutional layer only providing coarse localization information limit the performance upper bound of weakly-supervised tasks. Thus, we hope to obtain more fine-grained details to help locate the target objects better.

Fig. 1. Class activation maps of the horse category produced by Grad-CAM [2] (top row) and our LayerCAM (bottom row). The class activation maps are generated from $conv3\_3$ and $conv5\_3$ of VGG16 [10].

As the outputs of the shallow layers of CNN tend to have larger spatial resolutions, a natural way to acquire object details is to employ existing attention methods, such as Grad-CAM, to them. We show the class activation maps generated from $conv3\_3$ of VGG16 by Grad-CAM in Fig. 1. It can be seen that the localization becomes worse as the locations with strong values in the maps scatter around the whole image. We analyze Grad-CAM only considers capturing the global information of each feature map, where the local differences in it are lost. We have provided more discussion in Sec. II-A to analyze why Grad-CAM gets worse in shallow layers.

To generate reliable class activation maps for shallow layers to obtain more accurate fine-grained object localization information, we propose a simple yet effective method, Lay-

P.T. Jiang, C.B. Zhang, M.M. Cheng are with TKLNDST, CS, Nankai University. M.M. Cheng is the corresponding author (cmm@nankai.edu.cn).
* denotes equal contribution.
Q. Hou is with NUS.
Y. Wei is with UTS.

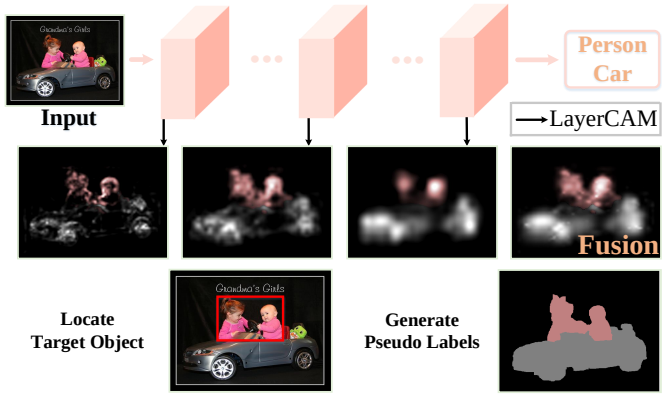


Fig. 2. An illustration of our LayerCAM. LayerCAM can be applied to any off-the-shelf CNN-based models and generates class activation maps from different layers. The fusion of class activation maps from different stages is beneficial for the object localization and semantic segmentation tasks.

erCAM, in this paper. Specifically, we rethink the relationships between the feature maps and their corresponding gradients. Unlike previous attention methods only considering the global information of each feature map, we utilize the gradients to highlight the different importance of each location in the feature map for the class of interest. Through such operation, the fine-grained details of the target objects can be effectively kept while the details in the background can be removed. Generally, LayerCAM offers the following advantages:

- LayerCAM can generate reliable class activation maps not only from the final convolutional layer but also from shallow layers, where we can obtain both coarse spatial locations and fine-grained object details.
- The class activation maps from different layers are often complementary. This advantage motivates us to combine them to generate more precise and integral class-specific object regions, which will significantly benefit weakly-supervised tasks.
- LayerCAM is easy to be applied to off-the-shelf CNN-based image classifiers without modifying the network architectures and the back-propagation way, making it more general and convenient to use.

To demonstrate the quality of the class activation maps, we apply them to the weakly-supervised object localization and semantic segmentation tasks. An illustration of our LayerCAM is shown in Fig. 2. Experiments on both tasks show that our approach achieves better object localization ability than previous attention methods, demonstrating the effectiveness of our LayerCAM. Besides, the property of generating fine-grained object locations from the shallow layers of CNN-based models can also be used to locate the tiny defects accurately in industrial images.

I. Related Work

A. Attention Methods

Researchers have proposed many attention methods [11], [12], [13], [14], [15] to locate the object regions of the class of interest from the powerful CNN based image classifiers [16], [17], [18], [19], [20], [21], [22]. The effectiveness of the attention methods [1], [2] or attention modules [23], [24], [25] benefit a lot of vision tasks [26], [27], [28], [29], [30], [31], [32]. Here, we mainly discuss two kinds of attention methods that are highly related to our work.

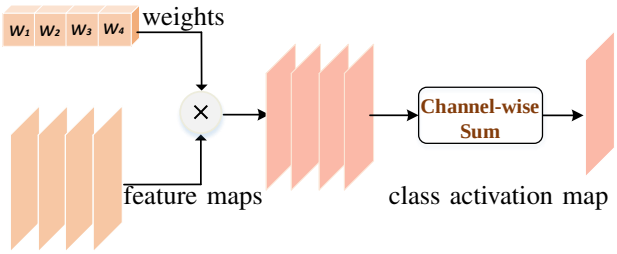


Fig. 3. The process of class activation mapping methods [1], [2], [3].

Class Activation Mapping. Such kinds of attention methods [1], [2], [3], [33] generate class activation maps from the final convolutional layer. We show the general procedures of these methods in Fig. 3. The class activation maps are obtained by multiplying each feature map by its weight and then performing a summation on all weighted feature maps. Finally, a ReLU operation is applied to filter out negative activations.

The difference between these attention methods is the way to generate the weight for each feature map. CAM [1] obtained the weights from the fully-connected layer. They replaced the first fully-connected layer in the image classifiers with a global average pooling layer. Grad-CAM [2] flowed the class-specific gradients to each feature map and then averaged the gradients of each feature map as its weight. In Grad-CAM++ [3], similar to [2], they also utilized the gradients of a feature map to generate its weight. Score-CAM [33] get rid of the dependence on gradients and generate the weight for each feature map through its forwarding score. One common point shared by the above methods is that they all generate reliable class activation maps from the final convolutional layer. Unlike these methods, our LayerCAM can generate reliable class activation maps from different layers of CNN.

Winner-take-all Scheme. Zhang et al. [34] propose a top-down back-propagation scheme, c-MWP, which passes signals in the network downwards based on a probability Winner-Take-All model. It can produce class activation maps for all convolutional layers of image classifiers. However, as verified in [2], the maps by c-MWP from the last convolutional layer are less faithful than Grad-CAM, which are rarely used in weakly-supervised tasks. Moreover, such a top-down process is complex and also needs more running time than Grad-CAM. Recently, PRM [35] utilizes local maximums, i.e., the peak values, as top signals to be back-propagated the network downwards in a winner-take-all manner to extract fine-detailed class instance activation maps.

Despite their ability to generate hierarchical class activation maps, the extracted cues from these maps often cover small object regions. Only the most relevant neurons are retained when the winner-take-all scheme is used. Besides, NormGrad [36] utilized identity layers to generate class activation maps for different layers. However, NormGrad is more likely to cap-

ture the small and discriminative object regions, not integral object regions. Different from them, the class activation maps generated by LayerCAM tend to cover more object regions than them. Additionally, LayerCAM is easy to be applied to off-the-shelf CNN-based image classifiers without modifying the network architectures, making the class activation maps are more easily available.

B. Hierarchical semantics

Many vision tasks, such as the challenging object detection task [37], [38], saliency object detection task [39], [40], and semantic segmentation task [41], [42], have benefited from the semantic knowledge of different feature hierarchies. Besides, Xie et al. [43] largely improved the edge detection task by utilizing the features from different hierarchies of CNNs. Wang et al. [44], [45] modeled the human parsing with a hierarchical structure. For visual object tracking, Shen et al. [46] made full use of the features of different hierarchies and fused them for better tracking results. Our LayerCAM also utilizes the hierarchical semantic knowledge from different layers. The class activation maps generated from shallow layers tend to capture the fine-grained details of the target objects. While the class activation maps generated from deep layers often locate coarse spatial object regions. The class activation maps from different hierarchies all help to locate the target objects.

C. Weakly-supervised object localization

Weakly-supervised object localization (WSOL) uses only image-level labels to find the tight boxes of the target objects. Some researchers [47], [48], [49], [50], [51], [52] attempt to solve WSOL as a multiple instance learning framework. Another kinds of methods [53], [54], [55], [56] selected the proper tight boxes for objects from the object proposal priors [57], [58].

Recently, a lot of WSOL methods [59], [60], [1], [8], [60], [61], [62], [63] utilizing attention methods have been proposed, such as CAM [1], Grad-CAM [2], and ACoL [8]. CAM and Grad-CAM identified the object regions by extracting the confident areas in class activation maps. However, the localization performance is limited because the confident areas are often small and coarse. Kim et al. [61] utilized two training steps to find different object localization information. Zhang et al. [8] used two CNN classification branches to find more confident areas from class activation maps based on the erasing strategy. The localization methods based on attention methods all generate class activation maps from the final convolutional layer of CNN. Different from previous localization methods, we attempt to mine more integral and accurate object locations by generating reliable class activation maps for different convolutional layers and finding more fine-grained localization information to help locate target objects accurately.

D. Weakly-supervised semantic segmentation

Weakly-supervised semantic segmentation (WSSS) with image-level labels has been widely studied because the image-level labels are easily available without much human effort. As image labels only provide the existence of a certain class without any spatial location information, this task is still a challenging problem. Despite the difficulty, a lot of WSSS works [64], [65], [66], [67], [68], [69], [70] have been proposed in the past years. Some works [71], [72], [73] utilize image labels to train segmentation models directly. Besides, because of the popularity of attention methods [1], [2], [3], many WSSS approaches [74], [75], [6], [76], [77], [78], [79], [80], [81] use the object localization cues extracted from the class activation maps. They first generate pseudo segmentation labels and then use them to train the segmentation models. The integrity of located objects largely affects the quality of pseudo segmentation labels. Our class activation maps fused from different layers of CNN can discover more integral and accurate object regions, benefiting the WSSS task.

E. Surface Defect Localization

Using computer-aided tools to check the quality of industrial products is a very important way to improve the quality and efficiency of industrial production. To find the surface defect locations in industrial images, many researchers [82], [83], [84], [85] often use fully supervised methods. Specifically, they need to annotate the location of the defects in industrial images and then train a segmentation or detection network. Although such kinds of methods achieve extraordinary performances, labeling the defects is quite difficult. Because the defects and their surrounding patches on the surface tend to have very low contrast, such as Fig. 7(a), labeling defects in industrial images becomes challenging.

Additionally, checking the defeats in industrial images in specific scenarios often requires professional knowledge, requiring lots of human effort and time. Thus, weakly-supervised methods worth studying as they can significantly reduce the annotation costs. We utilize class activation maps generated from shallow layers to locate tiny defects with various shapes in industrial images because the maps from shallow layers are sensitive to the fine-grained object details.

II. Methodology

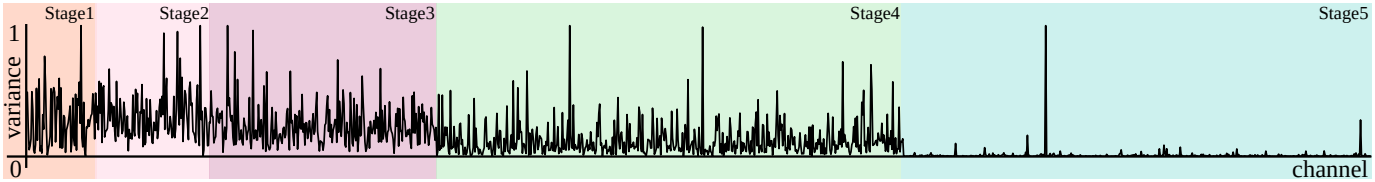
In this section, we first revisit the two most related approaches, i.e., Grad-CAM and GradCAM++. Then we introduce our method, LayerCAM.

A. Revisit Grad-CAM and Grad-CAM++

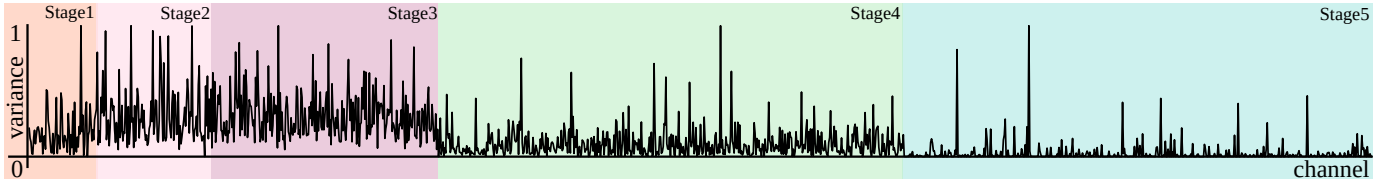
Formally, let f denote the image classifier and θ represent its parameters. For a given image I, when inputting I to the classifier, we can obtain the predicted score y^c of the target category c by

$$y^c = f^c(I, \theta). \tag{1}$$

Let A be the output feature maps of the final convolutional layer in CNN and A_k be the k-th feature map within A. The gradient of the prediction score y^c with respect to the



(a) GradCAM



(b) GradCAM++

(c) Feature maps

Fig. 4. (a-b) show the variances of the gradients corresponding to each feature map at different stages of VGG16. (c) illustrates the feature maps randomly selected from different stages.

spatial location (i, j) in the feature map A_k can be obtained by $g_{ij}^{kc} = \frac{\partial y^c}{\partial A_{ij}^k}$. To produce the class activation map of the target category c, Grad-CAM and Grad-CAM++ assign a *channel-wise weight* w_k to each feature map A_k. Then they perform a linear weighted summation on all feature maps in feature A. Finally, a ReLU operation is applied to remove the negative responses from the class activation map, which is formulated as

$$M^c = \text{ReLU}\left(\sum_k w_k^c \cdot A_k\right). \tag{2}$$

Grad-CAM obtains the *channel-wise weight* w_k^c for the feature map A_k by averaging the gradients of all locations in the feature map A_k, which is formulated as

$$w_k^c = \frac{1}{N}\sum_i \sum_j g_{ij}^{kc}, \tag{3}$$

where N denotes the number of locations in the feature map A_k. For Grad-CAM++ [3], the *channel-wise weight* w_k^c can be computed by

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} \cdot relu(g_{ij}^{kc}), \tag{4}$$

where α_{ij}^{kc} is computed by

$$\alpha_{ij}^{kc} = \frac{\left(g_{ij}^{kc}\right)^2}{2\left(g_{ij}^{kc}\right)^2 + \sum_a \sum_b A_{ab}^k (g_{ij}^{kc})^3}, \tag{5}$$

where (a, b) denotes the spatial location in A_k. The difference between Grad-CAM and Grad-CAM++ is that the latter utilizes both the feature maps and gradients to generate the *channel-wise weight*. Grad-CAM++ shows better object localization ability when multiple object instances occur.

Although Grad-CAM and Grad-CAM++ can generate reliable class activation maps from the final convolutional layer, the located object regions are often small and coarse. We hope to find more fine-grained localization information to remedy the class activation maps from the final convolutional layer to locate the target objects better. As we know, the shallow layers of CNN have larger spatial resolutions, causing them to capture more fine-grained details of the target objects. Thus, a natural idea to obtain fine-grained object details is to apply Grad-CAM or Grad-CAM++ to shallow layers. However, based on our experiments, the class activation maps from shallow layers produced by Grad-CAM or Grad-CAM++ often include many false positives, as shown in Fig. 1. In the following, we first analyze why Grad-CAM and Grad-CAM++ fail to generate reliable class activation maps for shallow layers and then introduce our method, LayerCAM.

Analysis Both Grad-CAM and Grad-CAM++ assign a global weight w_k^c to the k-th feature map A_k^c, where each location in A_k^c has the same weight w_k^c with each other. However, the feature maps in shallow layers tend to capture the fine-grained details, whether they belong to the target objects or background, as shown in Fig. 4(c). Thus, a global weight cannot eliminate the noisy regions in the background, which

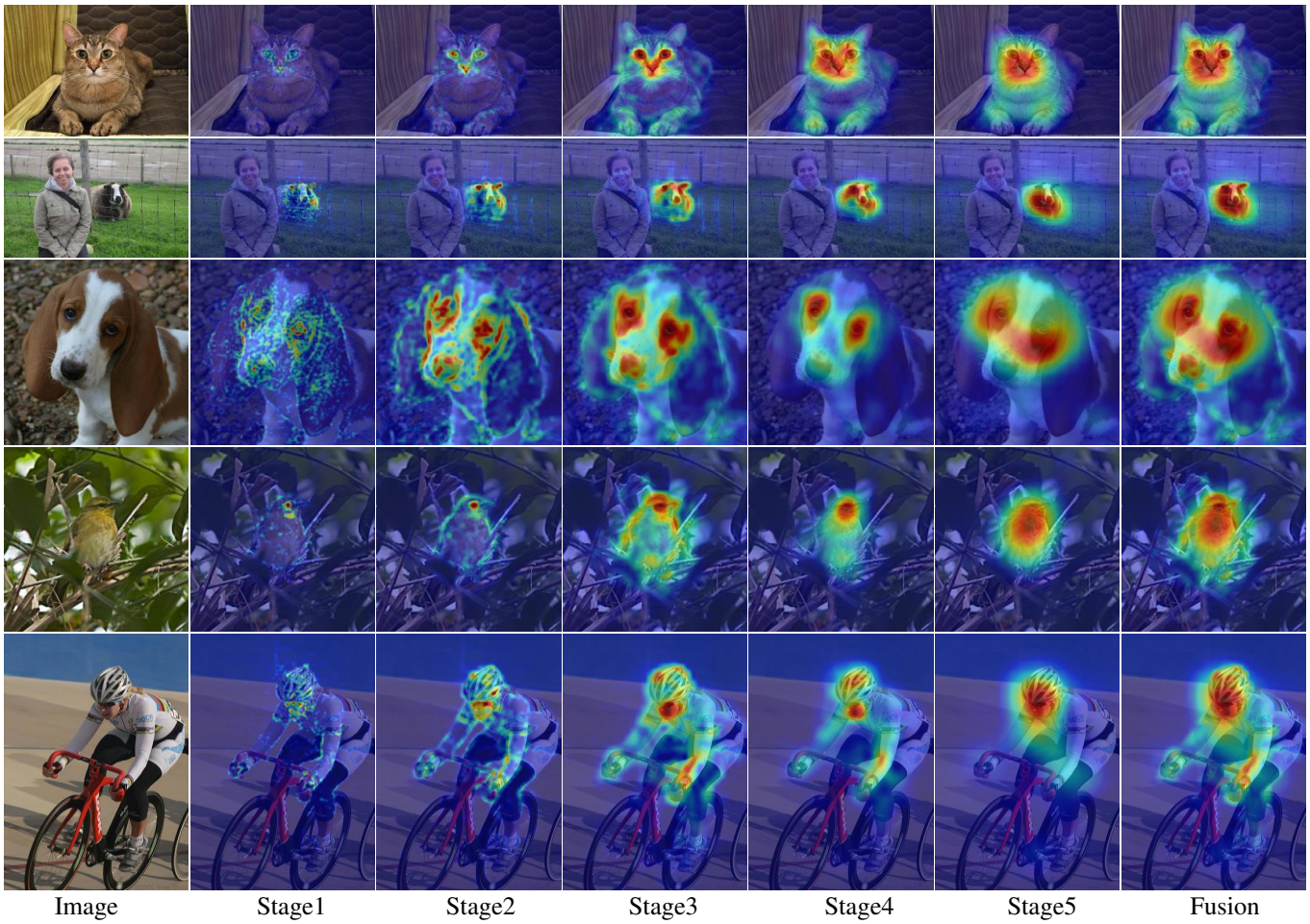


Fig. 5. The comparison of the class activation maps from different stages. The images are randomly selected from the PASCAL VOC dataset [86]. **Stage5** denotes the class activation maps are generated from the last convolutional layer of stage 5 in VGG16. **Fusion** denotes the class activation maps are fused from stage 3, stage 4, and stage 5. Notice that the class activation maps from stages 1, stage 2, and stage 3 are scaled according to Eqn. (9).

makes that the resulting class activation maps cannot locate target objects accurately.

Additionally, we also conduct a numerical analysis that whether the global weight can represent the importance of each location in one feature map. For Grad-CAM, we compute the variance of the gradients g^{kc}, where the variance denotes the difference of the gradient for each location to the average gradient, *i.e.,* w_k^c. And for Grad-CAM++, we compute the variance of $\alpha^{kc} \cdot relu(g^{kc})$ for the k-th feature map. We select the last convolutional layer of each stage from VGG16. As shown in Fig. 4(a-b), at the last stage, we can see that the variances corresponding to most feature maps tend to be zero. This demonstrates that the weight for each spatial location in the feature map is approximately equal to the global weight. Thus, at the last stage, both the global weights used by Grad-CAM and Grad-CAM++ can represent the importance of each spatial location in the feature map. However, at the shallow layers, the variances corresponding to most feature maps are very large. The global weight cannot represent the importance of different locations in the feature maps on the target category. Thus, Grad-CAM and Grad-CAM++ cannot generate reliable class activation maps for the shallow layers.

B. LayerCAM

Based on the above analysis, we propose LayerCAM, which enables the harvesting of reliable class activation maps for all layers in a very simple and effective manner. Specifically, to generate a separate weight for each spatial location in a feature map, we utilize the backward class-specific gradients. As empirically verified in [3], a positive gradient corresponding to a location in the feature map indicates that increasing the intensity of this location would have a positive influence on the prediction score of the target class. For the locations with positive gradients, we use their gradients as weights. Those locations with negative gradients are assigned with zero.

Formally, the weight of the spatial location (i, j) in the k-th feature map can be written as

$$w_{ij}^{kc} = relu(g_{ij}^{kc}). \tag{6}$$

To obtain the class activation map for a certain layer, LayerCAM first multiplies the activation value of each location in the feature map by a weight:

$$\hat{A}_{ij}^{k} = w_{ij}^{kc} \cdot A_{ij}^{k}. \tag{7}$$

Finally, the results $\hat{A}_k$ are linearly combined along the channel dimension to obtain the class activation map, which is formulated as follows:

$$M^c = \text{ReLU}\left(\sum_k \hat{A}^k\right). \tag{8}$$

Based on the above operation, the class activation maps generated from the shallow layers can capture reliable fine-grained object localization information, as shown in Fig. 5. We believe this mostly benefits from not only considering the importance of different channels, but also considering the importance of different spatial locations. The separate weight for each location can reflect the importance of different locations in the feature maps related to the target categories. We will give a more qualitative and quantitative analysis in the experiment section.

III. Experiments

In this section, the weakly-supervised object localization experiment is firstly conducted to verify the localization ability of LayerCAM. Then we utilize the image occlusion experiment to test the reliability of the general localization ability of class activation maps from the final convolutional layer. Moreover, we conduct the surface defect detection experiment to show the class activation maps from shallow layers can find fine-grained object localization information. Finally, we demonstrate that the combination of class activation maps from different stages is beneficial to the weakly-supervised semantic segmentation.

A. Weakly-supervised Object Localization

The object localization experiment is proposed in the ILSVRC benchmark [87], aimingat locating object bounding boxes for the top-predicted categories. We evaluate the localization ability of our method on the ILSVRC validation set that has 50000 images. The localization accuracy is measured by the *loc1* and *loc5* metrics. The *loc1* metric denotes that the estimated result falls into the correct category if the intersection over union (IoU) between the estimated bounding box and the ground-truth bounding box is greater than or equal to 0.5 and meanwhile the top 1 predicted class is correct. The *loc5* metric is for the top 5 predicted categories.

Implementation details. To generate object bounding boxes from class activation maps, we directly binarize them with the threshold of 15% of maximum intensity and then find the tight box of the largest connected segment as done in [1], [2]. We select the last convolutional layers of different stages in VGG-16 to generate class activation maps. For the class activation maps generated by LayerCAM from the *conv1_2* and *conv2_2* layer, the object locations with strong values in maps tend to scatter around the objects. Thus, following [11], we apply GraphCut [88] to generate a connected segmentation. Moreover, when combining the class activation maps from different layers, the class activation maps from the first three stages of VGG16 are scaled by Eqn. (9).

In Tab. I, we first show the localization ability of class activation maps from different stages of the VGG16. We find

TABLE I
COMPARISON OF THE LOCALIZATION ACCURACY OF THE CLASS ACTIVATION MAPS FROM DIFFERENT STAGES. THE 'S' IN THE FIRST ROW DENOTES 'STAGE' IN VGG16. **S5-S1** DENOTES THE LAST CONVOLUTIONAL LAYER OF EACH STAGE IN VGG16.

Method	Metric (%)	S5	S4	S3	S2	S1
Grad-CAM	*loc1*	43.62	18.32	8.87	19.59	13.95
	loc5	53.99	22.70	11.05	23.85	17.27
Grad-CAM++	*loc1*	45.44	41.11	35.33	31.70	31.32
	loc5	56.42	50.97	43.86	39.40	38.90
ScoreCAM	*loc1*	39.51	33.08	31.15	29.90	29.63
	loc5	49.63	41.63	39.30	37.80	37.46
NormGrad	*loc1*	38.94	40.85	38.67	32.05	29.94
	loc5	49.19	51.98	49.56	41.37	38.69
LayerCAM	*loc1*	**46.62**	**44.05**	**41.83**	**43.18**	**43.71**
	loc5	**57.83**	**55.02**	**52.28**	**53.60**	**54.34**

TABLE II
COMPARISON OF THE LOCALIZATION ACCURACY OF THE FUSED CLASS ACTIVATION MAPS FROM DIFFERENT STAGES. THE 'S' IN THE FIRST ROW DENOTES 'STAGE' IN VGG16. **S5-S1** DENOTES THE LAST CONVOLUTIONAL LAYER OF EACH STAGE IN VGG16.

Method	Metric (%)	S5	+S4	+S3	+S2	+S1
Grad-CAM	*loc1*	43.62	40.56	40.03	35.87	33.96
	loc5	53.99	50.11	49.47	44.48	42.17
Grad-CAM++	*loc1*	45.44	42.72	37.25	32.51	31.60
	loc5	56.42	52.95	46.19	40.40	39.23
ScoreCAM	*loc1*	39.51	37.26	31.88	29.94	29.52
	loc5	49.63	46.78	40.19	37.84	37.33
NormGrad	*loc1*	38.94	36.59	36.45	36.41	36.26
	loc5	49.19	46.02	45.86	45.79	45.59
LayerCAM	*loc1*	46.62	47.17	47.22	**47.24**	47.23
	loc5	57.83	58.67	58.72	**58.74**	58.74

the localization performance of LayerCAM outperforms Grad-CAM [2], Grad-CAM++ [3], ScoreCAM [33], and NormGrad [36] by a large margin, especially in shallow layers. This fact demonstrates LayerCAM can obtain more reliable fine-grained object localization information from shallow layers than those by Grad-CAM, Grad-CAM++, ScoreCAM, and NormGrad. As shown in Fig. 10(b-c), the class activation maps generated by Grad-CAM and Grad-CAM++ from the shallow layer cannot eliminate the noisy regions from the background and other categories. Our LayerCAM that assigns a separate weight for each location in the spatial dimension can consider the different importance to the class of interest, which can keep reliable object localization information while removing the background noise.

Additionally, we also show the localization performance of fusing class activation maps from different stages. For the class activation maps from shallow layers, the activation values are much lower than those from deep layers. When we don't use the scale function, the performance of the fused class activation maps will not be improved, as shown in Tab. IV (no scale). Thus, when combining the class activation maps from different layers, we first scale the class activation maps

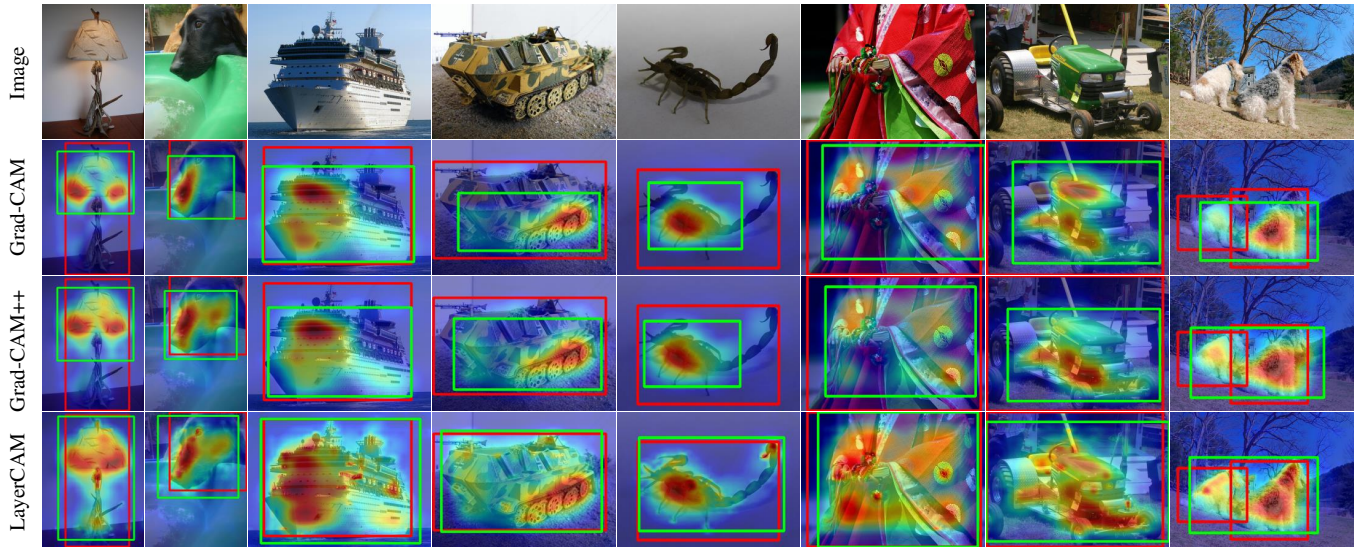


Fig. 6. Comparison of localization results among different methods. The images are randomly selected from the ILSVRC validation set [87]. The red box denotes the ground-truth box, and the green box denotes the predicted box. Our class activation maps fused from multiple layers can locate the object bounding-boxes more precisely than Grad-CAM and Grad-CAM++.

TABLE III
THE ABLATION OF γ.

Settings	Metric (%)	1	2	3	4
S5+S4+S3	*loc1*	47.18	**47.22**	47.17	47.04
	loc5	58.63	**58.72**	58.62	58.46
S5+S4+S3+S2	*loc1*	47.19	**47.24**	47.17	47.02
	loc5	58.63	**58.74**	58.63	58.46
S5+S4+S3+S2+S1	*loc1*	47.20	**47.23**	47.19	46.97
	loc5	58.65	**58.74**	58.64	58.39

TABLE IV
THE ABLATION OF DIFFERENT SCALE FUNCTIONS.

Settings	Metric	no scale	$tanh(x)$	$\sqrt[2]{x}$	$tan(x)$
S5+S4+S3	*loc1*	47.01	**47.18**	44.52	42.91
	loc5	58.40	**58.63**	55.62	53.39
S5+S4+S3+S2	*loc1*	47.00	**47.19**	44.53	40.07
	loc5	58.39	**58.63**	55.64	49.96
S5+S4+S3+S2+S1	*loc1*	46.91	**47.20**	44.51	38.67
	loc5	58.27	**58.65**	55.62	48.27

from shallow layers by a scale function, where the scaled maps are computed by

$$\hat{M}^c = \tanh\left(\frac{\gamma * M^c}{\max(M^c)}\right), \tag{9}$$

where γ is a scale factor. Then we utilize a simple element-wise maximum operation to combine the maps from different layers. It can be seen from Tab. III that LayerCAM achieves the best localization results when γ is set to 2. We also explore different kinds of scale functions, as shown in Tab. IV. When we use the $\sqrt[2]{x}$ scale function, the performance becomes much worse. This is because the $\sqrt[2]{x}$ scale function magnifies the value near 0 too much, which enhances the noise intensity. For example, 0.01 is scaled to 0.1. When we use the $tan(x)$ scale function, the performance also becomes much worse. The $tan(x)$ scale function scales the large values near 1 two much, which will restrain the magnification of the lower values after normalization. It can be seen that when using the $tanh(x)$ scale function, we can obtain better fusion results.

TABLE V
COMPARISON OF THE LOCALIZATION ACCURACY AMONG DIFFERENT METHODS. THE ATTENTION MAPS OF OTHER METHODS ARE ALL GENERATED FROM THE FINAL CONVOLUTIONAL LAYER. THE METHODS ASTERISK * DENOTES THE RESULTS ARE FROM THIS PAPER [2].

Methods	ACoL	ADL	CAM*	c-MWP*	Ours
loc1 (%)	45.83	44.92	42.80	29.08	**47.24**
loc5 (%)	59.43	-	54.86	36.96	**58.74**

As shown in Tab. II, the combination of class activation maps from different layers can gradually improve the localization performance. However, we have also observed that the performance gain becomes very small when gradually fusing the class activation maps from shallow layers. We analyze that the localization performance of the fused maps is limited due to the bounding box only indicates general object locations. It cannot measure the fine-grained details of the objects; for example, if the ear of the horse is found, the bounding box will not change much. In Sec. III-D, the segmentation results gradually increase when fusing the class activation maps from shallow layers, such as the class activation maps from stage 3, which can also verify the quality of the fused maps.

In Tab. V, we present the comparison of the localization performance among different methods. The attention methods in the rightmost two columns are all based on the original VGG16 model. The leftmost three localization methods all take the VGG16 architecture replacing the fully connected layer with the global average pooling layer. Compared to c-

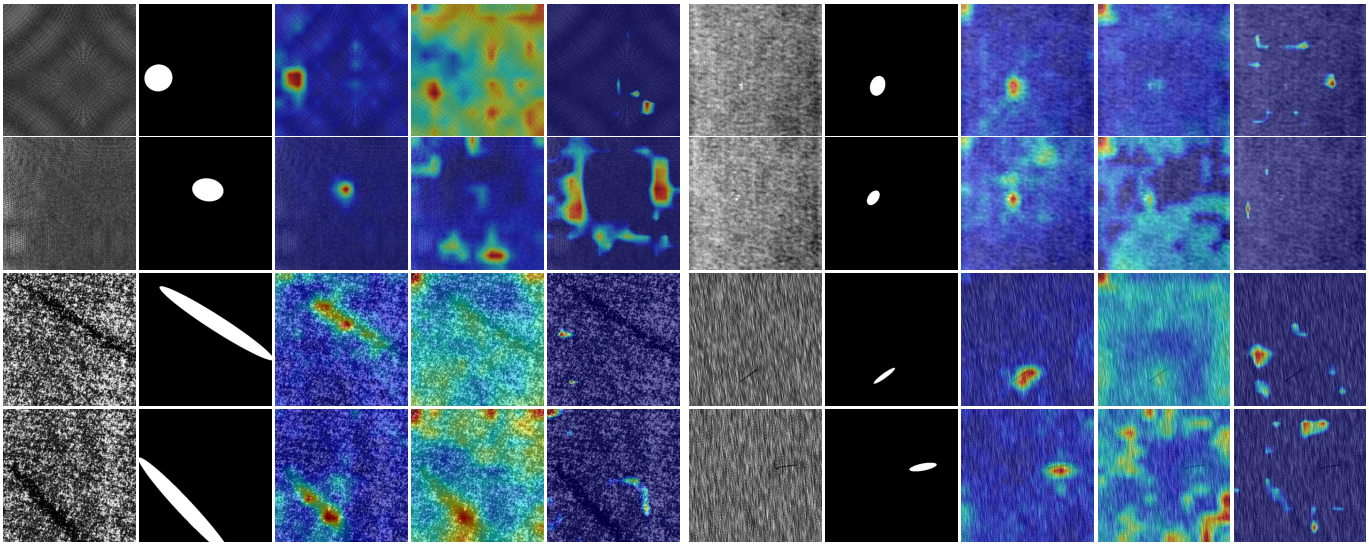


Fig. 7. The class activation map of industry defects. (a) Images. (b) Ground-truth. (c-e) shows class activation maps from *layer3* of ResNet50 generated by LayerCAM, Gard-CAM++, and Grad-CAM. The images are randomly selected from the DAGM-2007 defect dataset [89]. Our class activation maps can locate the object bounding boxes more precisely than Grad-CAM and Grad-CAM++. **Zoom in for the best view**.

MWP, CAM, Grad-CAM, and GradCAM++, it can be seen that our LayerCAM improves $loc1$ performance by 18.16%, 4.44%, 3.62%, and 1.80%, respectively. Our method also achieves a better result than some state-of-the-art localization approaches ACoL [8] and ADL [23]. They are specially designed to solve the object localization task. The comparison demonstrates the class activation maps by our LayerCAM can provide more reliable object localization information. The visual examples can be found in Fig. 6.

B. Image Occlusion

For the class activation maps generated from the final convolutional layer by LayerCAM, we conduct the image occlusion experiment proposed in [13] to verify the reliability of confidence regions indicated by these maps. The image occlusion experiment tests the importance of the occluded image regions to the final prediction. If the confidence regions are important, the predicted score of the target class would largely decrease when we input the occluded image. Specifically, on the ILSVRC validation set, we first select out the images that are predicted correctly by VGG16 used in Sec. III-A. For these truly predicted images, we occlude them with the threshold of 0.7 and then input them to the network. As shown in Tab. VI, we present the top-1 classification accuracy, top-5 classification accuracy, and the average predicted scores of the ground-truth category, respectively.

The performance of the image occlusion experiment is shown in Tab. VI. LayerCAM achieves a lower classification accuracy than Grad-CAM and Grad-CAM++, demonstrating that the class activation maps generated by LayerCAM from the final convolutional layer can discover more important spatial object regions for the target category. This experiment verifies the reliability of confidence regions located by our LayerCAM. When masking the images, we can find that removing the confidence regions by LayerCAM has more significantly reduced the predicted scores than Grad-CAM and Grad-CAM++.

TABLE VI
COMPARISON OF THE CLASSIFICATION ACCURACY ON THE IMAGE OCCLUSION EXPERIMENT. **CONFIDENCE**: DENOTES THE AVERAGE PREDICTED SCORES OF THE GROUND-TRUTH CATEGORY. LOWER IS BETTER.

Method	original	Grad-CAM	Grad-CAM++	LayerCAM
Top-1 Acc (%)	68.74	50.36	50.07	**48.26**
Top-5 Acc (%)	88.57	75.62	75.26	**73.43**
Confidence (%)	68.64	50.24	49.99	**48.12**

C. Industry Surface Defect Localization

For class activation maps generated from shallow layers of CNN, we utilize them to locate tiny defects with various shapes in industrial images. We regard this problem as a binary classification problem with or without defects in images. Then we use the image-level label to train a classifier based on ResNet50 [16]. Finally, we apply LayerCAM to locate defects in industrial images.

Implementation details. We do experiments on the DAGM-2007 defect dataset [89], which contains 3550 training images and 400 test images. This dataset contains many types of defects on different textured surfaces, as shown in Fig. 7(a-b). We train a defect image classifier on this dataset. We use SGD to optimize the classifier and train the model for 15 epochs with a batch size of 32. The initial learning rate is set as 0.001, and it decays at 5-th epoch and 10-th epoch, respectively. At the inference time, we apply our LayerCAM, Grad-CAM, and Grad-CAM++ to *layer3* of ResNet-50 to generate class activation maps, respectively. The generated maps are first

TABLE VII
COMPARISON OF DIFFERENT METHODS. THE SEGNET AND REFINENET ARE THE FULLY-SUPERVISED METHODS, WHILE THE OTHERS ARE WEAKLY-SUPERVISED METHODS. THE RESULTS ASTERISK * INDICATES THAT THEY ARE FROM THE ORIGINAL PAPER [82].

Methods	mIoU (%)	FPS
SegNet [90]	21.95*	17.92*
RefineNet [91]	**32.90***	31.05*
Grad-CAM	0.35	60.97
Grad-CAM++	6.46	60.24
LayerCAM	**27.26**	60.61

TABLE VIII
COMPARISON OF THE LOCALIZATION ACCURACY AMONG DIFFERENT LAYERS. **S4-S1** DENOTES *layer4-layer1* OF RESNET-50.

Setting	S4	S3	S2	S1	S4+S3	S3+S2
mIoU (%)	11.59	**27.26**	19.37	13.10	12.28	24.51

thresholded to a binary mask. Then we compute the IoU score among the binary mask and ground-truth mask. We search for the best threshold for each attention method and report their best performance. We also test the frames per second (FPS) of different methods. The running time is averaged over 100 iterations on an NVIDIA RTX 2080Ti.

We have shown the quantitative comparison among different methods in Tab. VII. The experimental results demonstrate that our method can locate defects more accurately than Grad-CAM and Grad-CAM++ while suppressing background noise. We have also shown several fully-supervised methods SegNet [90] and RefineNet [91] trained with pixel-level labels. Compared with them, our LayerCAM achieves comparable performance, but with about two times speed than them. Besides, we also show the qualitative comparison among different methods in Fig. 7. Compared with Grad-CAM and Grad-CAM++, LayerCAM can locate the defects with various kinds of shapes, while Grad-CAM and Grad-CAM++ cannot filter the interference information on the background.

In Tab. VIII, we have shown the localization accuracy of different layers. For the industry defect localization task, the performance of *layer3* is better than that of the fusion of multiple layers. This is because industry defects usually have small sizes and various shapes. As shown in Fig. 8, the low spatial resolution of class activation maps from *layer4* can only locate the defects coarsely, which cannot benefit the feature fusion. The class activation maps generated from *layer2* and *layer1* locate small defect regions with some noises. Thus, for the industry defect localization task, we only utilize the class activation maps from *layer3* instead of the multi-layer fusion.

D. Weakly-supervised segmentation

To further test the quality of our class activation maps, we apply them to the weakly-supervised semantic segmentation task that needs more pixel-accurate information. We utilize the class activation maps and the superpixels [92] to generate pseudo segmentation labels. As inspired by [35], we use the class activation maps as queries to collect object masks from the superpixels. We compute the probability of the existence of the category c by averaging the attention values in each superpixel,

$$S_c = \left(\frac{1}{|O|} \sum_{j \in O} M_j^c \right), \tag{10}$$

where O denotes a superpixel and M is the class activation map whose values are normalized to the range of [0, 1]. Then we select the maximum probability among all target categories and assign the corresponding category to all pixels in the superpixel. If the maximum probability is smaller than a fixed low threshold (in our experiment, the threshold is set to 0.3), the pixels in the superpixel are assigned with the background category. After assigning the semantic categories for each superpixel, we utilize them to constitute the pseudo segmentation labels to train a segmentation model.

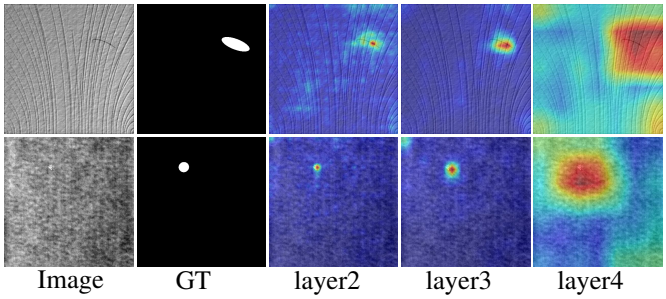


Fig. 8. The class activation maps from different layers on the defect localization task.

Implementation details. We perform the segmentation experiment on the popular PASCAL VOC 2012 dataset [86]. This dataset contains 20 semantic classes and the background. The original images are split into 1464 training images, 1449 validation images, and 1456 test images. Following the setting in [93], we utilize the augmented training set with 10,582 images to train the segmentation model and then compare our method with Grad-CAM and Grad-CAM++ on the validation and test sets. For ease of comparison, we use the CNN classifier as proposed in [1]. The last fully-connected (FC) layer with 1000 channels is modified to have 20 channels for the PASCAL VOC dataset. We adopt the VGG16 model pre-trained on ImageNet [87] to initialize our network and use the cross entropy loss to optimize it. During the inference time, we select the class activations maps generated from the last layers of each stage in VGG16. For Grad-CAM and Grad-CAM++, we only utilize the class activation maps from the final convolutional layer as the maps from shallow layers are much worse.

We adopt the popular Deeplab-LargeFOV [94] architecture based on VGG16 [10] as our segmentation network. We also train Deeplab-LargeFOV [94] based on ResNet [16] following the setting in [95]. The hyper-parameters for training the segmentation network are as follows: learning rate: *1e-*

Fig. 9. Examples of segmentation results produced by our method: (a) source images, (b) ground-truth, (c-e) segmentation results using class activation maps from stage 5, the combination of stage 5 and 4, and the combination of stage 5, 4, and 3, respectively.

TABLE IX
WEAKLY-SUPERVISED SEGMENTATION RESULTS ON THE PASCAL VOC DATASET. 'WEAK' MEANS THE APPROACHES WITH ONLY IMAGE-LEVEL SUPERVISION. FOR A FAIR COMPARISON, OUR APPROACH IS ALSO BASED ON THE DEEPLAB-LARGEFOV SEGMENTATION MODEL.

Methods	val (%)	test (%)
Grad-CAM	55.6	56.3
Grad-CAM++	55.5	56.1
LayerCAM (Ours, VGG16)	**60.8**	**61.4**
LayerCAM (Ours, ResNet101)	**63.0**	**64.5**

TABLE X
COMPARISONS OF THE MIOU SCORES ON THE PASCAL VOC VALIDATION SET WHEN COMBINING CLASS ACTIVATION MAPS FROM DIFFERENT STAGES.

S5	S4	S3	S2	S1	mIoU (%)
✔					55.6
	✔				55.0
		✔			50.8
			✔		50.5
				✔	46.0
✔	✔				57.1
✔	✔	✔			60.4
✔	✔	✔	✔		**60.8**
✔	✔	✔	✔	✔	60.2

3; learning rate policy: *poly*, batch size: *10*. We run the SGD for 16000 iterations. The learning rate decays at 12000 iterations by a factor of 10. At the inference time, we use the mean intersection-over-union (mIoU) metric to evaluate the segmentation results.

In Tab. IX, we report the performance of our method in terms of the mIoU scores. The performance of our method outperforms that of Grad-CAM and Grad-CAM++ by more than 5%. We also report the performance using the fusion of different stages of class activation maps, as shown in Tab. X. The mIoU score of our method using class activation maps from stage 5 of VGG16 is 55.6%. We observe that when continuously fusing the class activation maps from stage 4, stage 3, and stage 2 into stage 5 with the element-wise maximum operation, the mIoU score gradually increases (from 55.6% to 60.8%). This fact validates that our fused class activation maps can obtain more object localization information, which is beneficial for the segmentation task. Additionally, we also apply our fused class activation maps to a more advanced weakly-supervised semantic approach, DGCN [80]. As shown in Tab. XI, we can see that when replacing the CAM seeds with our LayerCAM seeds, the segmentation results can be further improved by about 3% mIoU score. The experimental results verify that the seeds generated by LayerCAM have better localization ability than those by CAM, which benefits the weakly-supervised semantic segmentation approach. As shown in Fig. 9, we show some qualitative segmentation results. It can be seen that fusing class activation maps from different stages of VGG16 can gradually increase the quality of the segmentation results. We notice that when fusing class activation maps of stage 1 into the final maps, the performance decreases from 60.8% to 60.2%. We analyze that the class activation maps from stage 1 lack class discrimination compared to those from other stages.

TABLE XI
COMPARISONS OF DGCN [80] WITH DIFFERENT CAM SEEDS.

Setting	val(%)	test (%)
DGCN-CAM	64.0	64.6
DGCN-LayerCAM	67.1	67.6

E. The influence of negative gradients

In Eqn. (6), LayerCAM utilizes ReLU to filter out the negative gradients. In this section, we explore the influence of the negative gradients. We first do experiments to study the impact of the negative gradients on the localization ability. As shown in Tab. XII, LayerCAM with negative gradients (LayerCAM-normal) achieves a much lower localization ability than LayerCAM. This fact verifies that the negative gradients in LayerCAM will decrease the localization ability. Additionally, we also measure the mIoU scores of fine-grained locations on the PASCAL VOC 2012 dataset. We generate the class activation maps for different stages of VGG16 and then

Fig. 10. Visualization of class activation maps. (a) Source images. (b-d) Class activation maps from the 'pool2' layer of VGG16 by Grad-CAM, Grad-CAM++, LayerCAM, and LayerCAM-normal, respectively. **LayerCAM-normal**: we use the original gradient of each location in the feature map as its weight.

TABLE XII
COMPARISON OF THE LOCALIZATION ACCURACY OF THE CLASS ACTIVATION MAPS FROM DIFFERENT STAGES. THE 'S' IN THE FIRST ROW DENOTES 'STAGE' IN VGG16. **S5-S1** DENOTES THE LAST CONVOLUTIONAL LAYER OF EACH STAGE IN VGG16.

Method	Metric	S5	S4	S3	S2	S1
LayerCAM-normal	*loc1*	42.09	37.63	34.74	34.12	30.86
	loc5	52.10	46.37	43.09	42.52	39.01
LayerCAM	*loc1*	46.62	44.05	41.83	43.18	43.71
	loc5	57.83	55.02	52.28	53.60	54.34

threshold them to binary masks by a hard threshold of 0.2. We compute the mIoU score between the thresholded mask and ground-truth mask.

In Tab. XIII, we present the mIoU scores of LayerCAM with different settings. It can be seen that LayerCAM utilizing the positive gradients as weights achieves higher mIoU scores than utilizing the normal gradients (with negative gradients), We also show the qualitative results from the *pool2* layer in Fig. 10(d-e). It can be seen that the class activation maps from LayerCAM with negative gradients lose many object localization information. Previous works [3], [12], [13] have also shown the importance of positive gradients in generating class activation maps or saliency maps. Thus, based on the empirical results, we filter out the negative gradients and select the positive gradient as the weight for each location in the feature map.

IV. CONCLUSION

In this paper, we propose an attention method, LayerCAM, which can generate reliable class activation maps from different layers of the CNN effectively. The class activation maps from deep layers can locate the general location of objects, and the maps from shallow layers can generate fine-grained object localization information. The combination of class activation maps from different layers can find more object locations, which is beneficial for improving the performance of the weakly-supervised tasks. Experiments show that LayerCAM has better object localization ability than current attention methods. Moreover, LayerCAM is easy to be utilized for any off-the-shelf CNN-based image classifiers without network architecture modification and changing the back-propagation way. Both PyTorch [96] and Jittor [97] versions of the source code will be made publically available.

TABLE XIII
THE COMPARISON OF THE CLASS ACTIVATION MAPS OF LAYERCAM WITH DIFFERENT SETTINGS. **LAYERCAM-NORMAL**: DENOTES WE USE THE ORIGINAL GRADIENT OF EACH LOCATION IN THE FEATURE MAP AS ITS WEIGHT.

mIoU(%)	S5	S4	S3	S2	S1
LayerCAM-normal	34.3	22.2	14.4	8.4	4.8
LayerCAM	36.2	35.7	31.5	21.8	11.1

ACKNOWLEDGMENTS

This research was supported by the National Key Research and Development Program of China under Grant No. 2018AAA0100400, NSFC (61922046), S&T innovation project from Chinese Ministry of Education, and the Fundamental Research Funds for the Central Universities (Nankai University, NO. 63213090).

REFERENCES

[1] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.
[2] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, 2020.
[3] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 839–847.
[4] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
[5] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4981–4990.
[6] Q. Hou, P. Jiang, Y. Wei, and M.-M. Cheng, "Self-erasing network for integral object attention," in *Adv. Neural Inform. Process. Syst.*, 2018, pp. 549–559.
[7] S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han, "Weakly supervised semantic segmentation using web-crawled videos," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 3626–3635.
[8] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. Huang, "Adversarial complementary learning for weakly supervised object localization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
[9] D. Li, J. Huang, Y. Li, S. Wang, and M. Yang, "Progressive representation adaptation for weakly supervised object localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1424–1438, 2020.
[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.
[11] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Int. Conf. Learn. Represent.*, 2014.
[12] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," in *Int. Conf. Learn. Represent. Worksh.*, 2015.

[13] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[14] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable basis decomposition for visual explanation," in *Eur. Conf. Comput. Vis.*, 2018, pp. 119–134.

[15] B. N. Patro, M. Lunayach, S. Patel, and V. P. Namboodiri, "U-cam: Visual explanation using uncertainty based class activation maps," in *Int. Conf. Comput. Vis.*, 2019, pp. 7444–7453.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[17] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2021.

[18] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1492–1500.

[19] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Brit. Mach. Vis. Conf.*, 2016.

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4700–4708.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2818–2826.

[22] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 10 096–10 105.

[23] J. Choe and H. Shim, "Attention-based dropout layer for weakly supervised object localization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2219–2228.

[24] W. Wang, J. Shen, and H. Ling, "A deep network solution for attention and aesthetics aware photo cropping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1531–1544, 2018.

[25] W. Wang, J. Shen, X. Lu, S. C. Hoi, and H. Ling, "Paying attention to video object pattern understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[26] S. Yang, Y. Kim, Y. Kim, and C. Kim, "Combinational class activation maps for weakly supervised object localization," in *IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2941–2949.

[27] Z. Shi, T. M. Hospedales, and T. Xiang, "Bayesian joint modelling for object localisation in weakly labelled images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 1959–1972, 2015.

[28] Y. Chen, Y. Lin, M. Yang, and J. Huang, "Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[29] Q. Meng, W. Wang, T. Zhou, J. Shen, L. Van Gool, and D. Dai, "Weakly supervised 3d object detection from lidar point cloud," in *Eur. Conf. Comput. Vis.*, 2020, pp. 515–531.

[30] Q. Hou, D. Massiceti, P. K. Dokania, Y. Wei, M.-M. Cheng, and P. H. Torr, "Bottom-up top-down cues for weakly-supervised semantic segmentation," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2017, pp. 263–277.

[31] Q. Meng, W. Wang, T. Zhou, J. Shen, Y. Jia, and L. Van Gool, "Towards a weakly supervised framework for 3d point cloud object detection and annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[32] H. Cholakkal, G. Sun, F. S. Khan, and L. Shao, "Object counting and instance segmentation with image-level supervision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 12 397–12 405.

[33] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2020, pp. 24–25.

[34] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," in *Eur. Conf. Comput. Vis.*, 2016.

[35] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, "Weakly supervised instance segmentation using class peak response," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3791–3800.

[36] S.-A. Rebuffi, R. Fong, X. Ji, and A. Vedaldi, "There and back again: Revisiting backpropagation saliency methods," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 8839–8848.

[37] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2117–2125.

[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[39] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 678–686.

[40] W. Wang, S. Zhao, J. Shen, S. C. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 1448–1457.

[41] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[42] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1925–1934.

[43] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.

[44] W. Wang, T. Zhou, S. Qi, J. Shen, and S.-C. Zhu, "Hierarchical human semantic parsing with comprehensive part-relation modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[45] W. Wang, H. Zhu, J. Dai, Y. Pang, J. Shen, and L. Shao, "Hierarchical human parsing with typed part-relation reasoning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 8929–8939.

[46] J. Shen, X. Tang, X. Dong, and L. Shao, "Visual object tracking by hierarchical attention siamese network," *IEEE transactions on cybernetics*, vol. 50, no. 7, pp. 3068–3080, 2019.

[47] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3512–3520.

[48] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 189–203, 2016.

[49] R. Gokberk Cinbis, J. Verbeek, and C. Schmid, "Multi-fold mil training for weakly supervised object localization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2409–2416.

[50] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie, "Weakly supervised object localization with stable segmentations," in *Eur. Conf. Comput. Vis.*, 2008, pp. 193–207.

[51] V. Kantorov, M. Oquab, M. Cho, and I. Laptev, "Contextlocnet: Context-aware deep network models for weakly supervised localization," in *Eur. Conf. Comput. Vis.*, 2016, pp. 350–365.

[52] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, and Q. Ye, "C-mil: Continuation multiple instance learning for weakly supervised object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2199–2208.

[53] Y. Zhu, Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, "Soft proposal networks for weakly supervised object localization," in *Int. Conf. Comput. Vis.*, 2017, pp. 1841–1850.

[54] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu, "Deep self-taught learning for weakly supervised object localization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1377–1385.

[55] E. W. Teh, M. Rochan, and Y. Wang, "Attention networks for weakly supervised object localization." in *Brit. Mach. Vis. Conf.*, 2016, pp. 1–11.

[56] W. Xu, Y. Wu, W. Ma, and G. Wang, "Adaptively denoising proposal collection for weakly supervised object localization," *Neural Processing Letters*, vol. 51, no. 1, pp. 993–1006, 2020.

[57] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 128–140, 2016.

[58] Y. Liu, P.-T. Jiang, V. Petrosyan, S.-J. Li, J. Bian, L. Zhang, and M.-M. Cheng, "Del: Deep embedding learning for efficient image segmentation." in *Int. Joint Conf. Artif. Intell.*, 2018.

[59] C. Wang, K. Huang, W. Ren, J. Zhang, and S. Maybank, "Large-scale weakly supervised object localization via latent category learning," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1371–1385, 2015.

[60] X. Zhang, Y. Wei, G. Kang, Y. Yang, and T. Huang, "Self-produced guidance for weakly-supervised object localization," in *Eur. Conf. Comput. Vis.*, 2018, pp. 597–613.

[61] D. Kim, D. Cho, D. Yoo, and I. So Kweon, "Two-phase learning for weakly supervised object localization," in *Int. Conf. Comput. Vis.*, 2017, pp. 3534–3543.

[62] H. Xue, C. Liu, F. Wan, J. Jiao, X. Ji, and Q. Ye, "Danet: Divergent activation for weakly supervised object localization," in *Int. Conf. Comput. Vis.*, 2019, pp. 6589–6598.

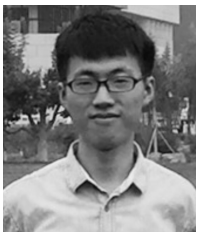
[63] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *Int. Conf. Comput. Vis.* IEEE, 2017, pp. 3544–3553.

[64] L. Zhang, Y. Gao, Y. Xia, K. Lu, J. Shen, and R. Ji, "Representative discovery of structure cues for weakly-supervised image segmentation," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 470–479, 2014.

[65] Z. Shi, Y. Yang, T. M. Hospedales, and T. Xiang, "Weakly-supervised image annotation and segmentation with objects and attributes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2525–2538, 2017.
[66] X. Li, H. Ma, and X. Luo, "Weaklier supervised semantic segmentation with only one image level annotation per category," *IEEE Trans. Image Process.*, vol. 29, pp. 128–141, 2020.
[67] C. Redondo-Cabrera, M. Baptista-Ríos, and R. J. López-Sastre, "Learning to exploit the prior network knowledge for weakly supervised semantic segmentation," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3649–3661, 2019.
[68] L. Zhang, Y. Yang, Y. Gao, Y. Yu, C. Wang, and X. Li, "A probabilistic associative model for segmenting weakly supervised images," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4150–4159, 2014.
[69] L. Jing, Y. Chen, and Y. Tian, "Coarse-to-fine semantic segmentation from image-level labels," *IEEE Trans. Image Process.*, vol. 29, pp. 225–236, 2020.
[70] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao, "Learning from weak and noisy labels for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 486–500, 2017.
[71] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a dcnn for semantic image segmentation," in *Int. Conf. Comput. Vis.*, 2015.
[72] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *ICCV*, 2015, pp. 1796–1804.
[73] N. Araslanov and S. Roth, "Single-stage semantic segmentation from image labels," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 4253–4262.
[74] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[75] A. Roy and S. Todorovic, "Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[76] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
[77] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7014–7023.
[78] P.-T. Jiang, Q. Hou, Y. Cao, M.-M. Cheng, Y. Wei, and H.-K. Xiong, "Integral object mining via online attention accumulation," in *Int. Conf. Comput. Vis.*, 2019, pp. 2070–2079.
[79] G. Sun, W. Wang, J. Dai, and L. Van Gool, "Mining cross-image semantics for weakly supervised semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 347–365.
[80] J. Feng, X. Wang, and W. Liu, "Deep graph cut network for weakly-supervised semantic segmentation," *Science China Information Sciences*, vol. 64, no. 3, p. 130105, 2021.
[81] X. Li, T. Zhou, J. Li, Y. Zhou, and Z. Zhang, "Group-wise semantic mining for weakly supervised semantic segmentation," *arXiv preprint arXiv:2012.05007*, 2020.
[82] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "Pga-net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE TII*, 2019.
[83] B. Su, H. y. Chen, P. Chen, G. Bian, k. Liu, and W. Liu, "Deep learning-based solar-cell manufacturing defect detection with complementary attention network," *IEEE TII*, 2020.
[84] Z. Tang, E. Tian, Y. Wang, L. Wang, and T. Yang, "Non-destructive defect detection in castings by using spatial attention bilinear convolutional neural network," *IEEE TII*, 2020.
[85] S. Lu, J. Feng, H. Zhang, J. Liu, and Z. Wu, "An estimation method of defect size from mfl image using visual transformation convolutional neural network," *IEEE TII*, vol. 15, no. 1, pp. 213–224, 2019.
[86] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, 2015.
[87] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, 2015.
[88] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Int. Conf. Comput. Vis.*, vol. 1. IEEE, 2001, pp. 105–112.
[89] M. Wieler and T. Hahn, "Weakly supervised learning for industrial optical inspection," 2007.
[90] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.
[91] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[92] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.
[93] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Int. Conf. Comput. Vis.*, 2011.
[94] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.
[95] A. Chaudhry, P. K. Dokania, and P. H. Torr, "Discovering class-specific pixels for weakly-supervised semantic segmentation," *Brit. Mach. Vis. Conf.*, 2017.
[96] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inform. Process. Syst.*, 2019.
[97] S.-M. Hu, D. Liang, G.-Y. Yang, G.-W. Yang, and W.-Y. Zhou, "Jittor: a novel deep learning framework with meta-operators and unified graph execution," *Science China Information Sciences*, vol. 63, no. 12, pp. 1–21, 2020.

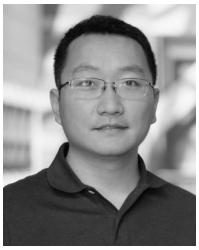
Peng-Tao Jiang is a Ph.D. student from the College of Computer Science at Nankai University, under the supervision of Prof. Ming-Ming Cheng. Before that, he received the bachelor degree from Xidian university in 2017. His research interests include weakly-supervised tasks and model interpretability.

Chang-Bin Zhang is a master student from the College of Computer Science at Nankai University, under the supervision of Prof. Ming-Ming Cheng. Before that, he received the bachelor degree from China University of Mining and Technology in 2019. His research interests include deep learning and computer vision.



Qibin Hou received his Ph.D. degree from School of Computer Science, Nankai University, under the supervision of Prof. Ming-Ming Cheng. Currently, he is a research fellow working with Prof. Jiashi Feng at National University of Singapore. His research interests include deep learning, image processing, and computer vision.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did 2 years research fellow, with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests includes computer graphics, computer vision, and image processing. He received research awards including ACM China Rising Star Award, IBM Global SUR Award, CCF-Intel Young Faculty Researcher Program. He is on the editorial boards of IEEE TIP.



Yunchao Wei is currently an Assistant Professor at the University of Technology Sydney. He received his PhD degree from Beijing Jiaotong University in 2016. Before joining UTS, he was a Postdoc Researcher in Prof. Thomas Huang's Image Formation and Professing (IFP) group at Beckman Institute, UIUC, from 2017 to 2019. His research interests mainly include Deep learning and its applications in computer vision, e.g., image classification, learning with imperfect data.