Segmentation and Shape Extraction from Convolutional Neural Networks

Mai Lan Ha*, Gianni Franchi*, Michael Möller, Andreas Kolb, Volker Blanz
University of Siegen, Germany
{hamailan, blanz}@informatik.uni-siegen.de
{gianni.franchi, michael.moeller, andreas.kolb}@uni-siegen.de

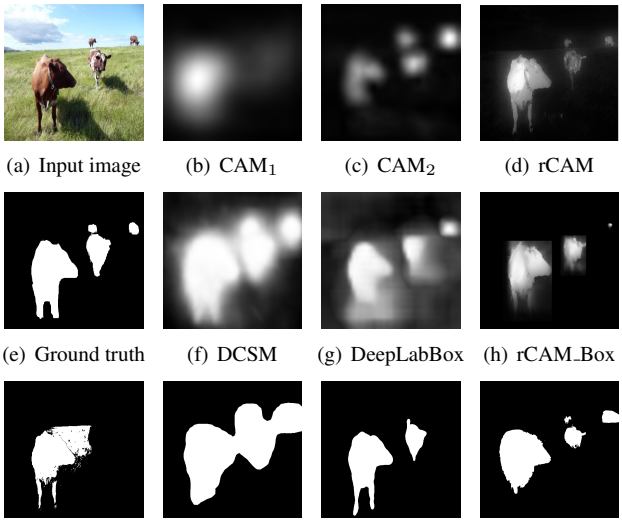
Abstract

We propose a novel method for creating high-resolution class activation maps from a given deep convolutional neural network which was trained for image classification. The resulting class activation maps not only provide information about the localization of the main objects and their instances in the image, but are also accurate enough to predict their shapes. Rather than pursuing a weakly supervised learning strategy, the proposed algorithm is a multi-scale extension of the classical class activation maps using a principal component analysis of the classification network feature maps, guided filtering, and a conditional random field. Nevertheless, the resulting shape information is competitive with state-of-the-art weakly supervised segmentation methods on datasets on which the latter have been trained, while being significantly better at generalizing to other datasets and unknown classes.

1. Introduction

The era of Deep Convolutional Neural Networks (DCNNs) has led to impressive advances on the problem of *image classification*. The improvements in the network architectures, for example in AlexNet [16], VGG [29], or GoogLeNet [30], as well as the training of deeper models were made possible by the availability of extremely large-scale datasets such as ImageNet [8] in which images are annotated with labels.

On the contrary, there is a limitation in creating big datasets for learning-based approaches to image segmentation. Such datasets require a pixel-accurate labeling of thousands of images by human observers. This is the reason why researchers have turned their attention to *weakly supervised segmentation methods* such as [4, 6, 22, 17, 5, 34] that take advantage of training on labeled images without any localization information. The goal is still to provide an accurate segmentation without entirely relying on the availability of large-scale segmentation datasets.



(a) Input image (b) CAM_1 (c) CAM_2 (d) rCAM
(e) Ground truth (f) DCSM (g) DeepLabBox (h) rCAM_Box
(i) CCNN (j) SEC (k) TransferNet (l) rCAM binary

Figure 1. Illustrating the behavior of the proposed rCAM based shape extraction: To detect the main objects in 1(a), the classical CAM method [34] provides the rough location of the foreground cow in 1(b). Smaller cows are located in 1(c) when the input image is upsampled. However, neither 1(b) or 1(c) is accurate enough to provide objects' shape. Our extensions in 1(d) and 1(h) provide segmentations that are at least as detailed as the competing methods [27] and [14] shown in 1(f) and 1(g), respectively, while generalizing well over a wider variety of different datasets. Our rCAM binarized version in 1(l) also shows better result than those methods that produce binary segmentation maps such as [23], [15] and [12] in 1(i), 1(j) and 1(k) respectively.

Zhou *et al*. showed in [34] that some localization information about the main object, i.e., the object with the highest classification score, can be extracted from a DCNN that had only been trained on image classification. Their technique is based on computing a class activation map (CAM) which identifies those regions in an image that lead the classification network to make a certain prediction about the image label.

Our work goes a step further by providing a high-

*The two authors contributed equally to this work.

resolution CAM that not only localizes all the instances of the main object in the image and but also provides shape information which is accurate enough to be used for image segmentation without requiring any additional training (see Figure 1). Opposed to the original CAM method [34], the proposed method is able to locate the whole object body rather than only discriminative regions which often cover only parts of the objects. For example, the heads of animals are the most discriminative parts, and can be effectively used to classify different animals. However, we aim to discover the whole animal body rather than just its head.

Note that our method still differs from *semantic image segmentation*, where every pixel in an image is classified, and from *object segmentation*, where all the objects in an image are segmented. The proposed method segments all instances of the main object in an image only. To bridge the gap between our method and the segmentation methods, we apply our high resolution CAM algorithm on region proposals produced by Faster-RCNN [25]. In either case, our method is comparable to the state-of-the-art weakly supervised segmentation methods which are intensively evaluated in Section 4. Although our method does not contain any fine-tune training stage of the classification network, it performs favorable in comparison to previous CAM methods as well as to state-of-the-art weakly supervised segmentation methods, particularly with respect to the ability to generalize across different datasets.

Our proposed method can be summarized in four steps: (i) Firstly, we create two CAMs at different scales from two different resolutions of the input image using the GoogLeNet-GAP network [34]. (ii) We extract the shape information from GoogLeNet-GAP using a principal component analysis (PCA) on a particular set of response maps. (iii) The two CAMs are upsampled by the guided filter [11] that uses the extracted shape information. (iv) The upsampled CAMs are merged to create a high-resolution class activation map. Finally, we use the Conditional Random Field in [32] to improve the accuracy of the shape prediction.

2. Related work

The difficulty of creating large-scale image segmentation datasets for training deep neural networks on one hand and the urgent need to extract localization and shape information from images on the other hand have sparked two lines of research, namely localization and weakly supervised segmentation. CAM methods, which are a subset of localization methods, try to localize objects by identifying pixels that activate the class of interest. Alternatively, weakly supervised segmentation techniques use different constraints and information that is less than segmentation ground truth to train or fine-tune DCNNs to perform segmentation tasks. Our article falls in between these two types of approaches.

Understanding DCNNs and Class Activation Maps In order to have a better understanding of the image classification process, various works identify the most important pixels used by a DCNN to classify an image. Bazzani *et al*. [4] apply masks at different locations on an image and classify each result. They study the link between the positions of the masks and the classification scores to localize objects. Simonyan *et al*. [28] predict a heat map by altering the input image. Oquab *et al*. [22] use a particular DCNN composed of a fully convolutional network which outputs K images, where K is the number of classes, followed by a global max pooling (GMP) and then a fully connected layer. Thanks to the K images before the fully connected layer, Oquab *et al*. localize the pixels that activate the class. Similarly, Zhou *et al*. [34] proposed a DCNN architecture, illustrated in Figure 2(a), that is able to classify an image. While their architecture is similar to GoogLeNet [30], a global average pooling (GAP) followed by a fully connected layer is used after the fully convolutional network. According to [34], GAP provides better localization results than GMP. Selvaraju *et al*. [26] propose a technique to extract the discriminative pixels based on the gradient of a DCNN. Based on CAMs produced by Zhou *et al*. [34], Wei *et al*. proposed an adversarial erasing method to iteratively expand the discriminative object regions [31]. Their mined regions are then used to train semantic segmentation. All the above techniques aim to localize the most important pixels used by a DCNN to classify an image. However they can only provide very crude estimations of the objects' shape.

Weakly supervised object segmentation Recent works [21, 14, 15, 23, 12, 27] have explored weakly-supervised object segmentation. While weakly supervised learning algorithms do not have access to the complete (semantic) segmentation of the training images, they vary strongly in the amount and detail of information of the training data. [23, 27, 15, 21] use image class labels only, which provide information about which objects are present in each image, but do not contain any localization information. More information can be exploited via bounding boxed as for instance in [14]. [21, 12, 27] learn shape information from other databases to improve semantic segmentation results. Other techniques like [15, 23] add some constraints on the shape of the objects. These constraints are used as a prior in order to improve the segmentation results.

3. High-resolution Class Activation Maps (rCAMs)

In this section, we present a method for producing high-resolution class activation maps (rCAMs) that not only localize the main object in an image but also predict its shape accurately. The proposed method is based on extracting

shapes from the GoogLeNet-GAP network [34] and using such information together with multi-scale CAMs to increase their resolution. The processes and overall structure of the framework are illustrated in Figure 2. It consists of extracting CAMs and shapes at two different scales, using the shape information for an upsampling of the activation maps, and finally fusing and refining the latter to obtain the rCAM result. In the following subsections, we will detail each of these steps.

3.1. Multi-scale CAMs extraction

We use the GoogLeNet-GAP network to create CAMs [34] as the basic components for constructing rCAM. The GoogLeNet-GAP mainly consists of convolutional layers. After the last convolutional layer, a Global Average Pooling (GAP) is performed and the GAP results are fed into a fully connected layer for the final classification producing a 1000-dimensional vector denoted P which holds the class probabilities for the classification result. Let us denote C_{CAM} the set of response maps of the CAM layer and w_{ij} the fully connected weight connecting the response map i (denoted C^i_{CAM}) and the coordinate j of P. The CAM of the class j at the position x is defined in [34] as: $\text{CAM}(x)^j = \sum_{i=1}^{N} w_{ij} C^i_{\text{CAM}}(x)$, where N is the number of response maps of the CAM layer. For an input image I of size 224×224, GoogLeNet-GAP produces CAM of size 14×14 that localizes the first object with the highest classifaction probability (Figure 2(a)).

Instead of using a single scale we resize every input image to images I_1 of size 224×224 and I_2 of size 448×448 by bilinear interpolation. The images I_1 and I_2 are feedforwarded to GoogLeNet-GAP to generate CAM_1 of size 14×14 and CAM_2 of size 28×28, respectively (Figure 2(a)). We discover that while CAM_1 provides the coarse discriminative regions for the main object, CAM_2 gives us finer discriminative regions that are sometimes overlooked by CAM_1 (see Figure 4 for an example).

The usage of the image I_2 of size 448×448 creates a zoom-out effect. The dominance of the discriminative regions discovered in the image I_1 of size 224×224 is reduced and the finer discriminative regions have an opportunity to be discovered in the image I_2. According to our experiments, CAM_2 is especially useful when there are multiple instances of the main object, for example many cows in the image in Figure 1. On the other hand, CAM_1 is very important for the classification and localization of the main object due to the suppression of small objects. Therefore, CAM_1 and CAM_2 do not compete but complement each other.

3.2. Shape extraction from GoogLeNet-GAP

Traditionally, object recognition or shape estimation uses hand-crafted features such as SIFT [20], or descriptors like the color, texture, or gradient of an image. The robustness of a method is based on the invariance of such features to factors such as scale, illumination, or rotation. However in DCNNs, one does not need to define features. Instead, the features are learnt and embedded inside DCNNs for us to discover [10, 33].

A DCNN can be divided into two parts. The first part involves a set of layers that form a Fully Convolutional Network (FCN). Each layer in FCN contains a series of convolutional operations followed by non-linear operators such as activation and pooling. The second part consists of Fully Connected Layers (FCL) that lead to the classification results. We focus on the FCN of GoogLeNet-GAP. The output of each convolution kernel in the FCN is an image called response map. Our goal is to find a set of response maps that contain shape information and extract the shape.

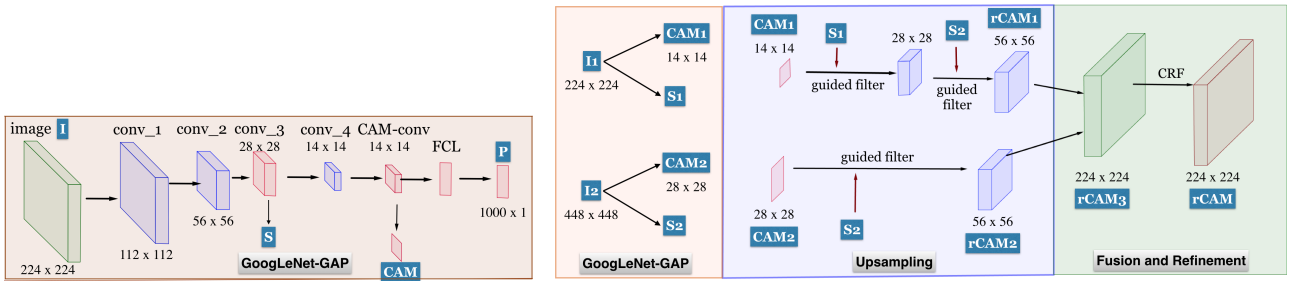
The FCN of the GoogLeNet-GAP architecture is a concatenation of convolution and pooling layers: for an input image I_1 of size 224×224, it produces response maps of sizes 112×112, 56×56, 28×28 and 14×14. By gathering all these response maps into four groups according to their sizes, we have four sets of response maps C^l with $l \in \{112, 56, 28, 14\}$. Each C^l is a cubic tensor such that $C^l \in \mathbb{R}^{l \times l \times D_l}$, where D_l is the number of response maps of size $l \times l$. Therefore, C^l can be decomposed into l^2 vectors v_k where $v_k \in \mathbb{R}^{D_l}$ and $k \in [\![1, l^2]\!]$.

To condense the information of the feature maps C^l, we apply a Principal Component Analysis (PCA) [13] to reduce the dimension of v_k from D_l to 3 by extracting the first three components, mapping $C^l = \{v_k\}_{k=1}^{l^2} \subset \mathbb{R}^{D_l} \longmapsto \tilde{C}^l = \{\tilde{v}_k\}_{k=1}^{l^2} \subset \mathbb{R}^3$. The resulting principal components represent the response maps C^l by more compact sets $\tilde{C}^l$ and yield a better understanding of the information contained in each of the feature maps, see Figure 3.

In order to discover the features of the response maps, we built a small dataset that is composed of 200 binary shape images and performed color and texture transformations on these shapes. We studied how the response maps change when the color and texture information varies. According to our numerical experiments $\tilde{C}^{112}$ and $\tilde{C}^{56}$ contain mainly gradient information, $\tilde{C}^{28}$ provides shape structures, and $\tilde{C}^{14}$ yields a heat map revealing the location of the main objection. This leads us to define $S_1 := \tilde{C}^{28}$ to be a shape representation of the input image I_1.

By feeding an input image I_2 of size 448×448 into the network and performing a PCA of the feature maps, one again obtains four compact response maps whose resolution is four times larger than the resolution of the corresponding feature maps of I_1. Again, the shape information S_2 is defined as the compact response map of the third layer such that $S_2 \in \mathbb{R}^{56 \times 56 \times 3}$.

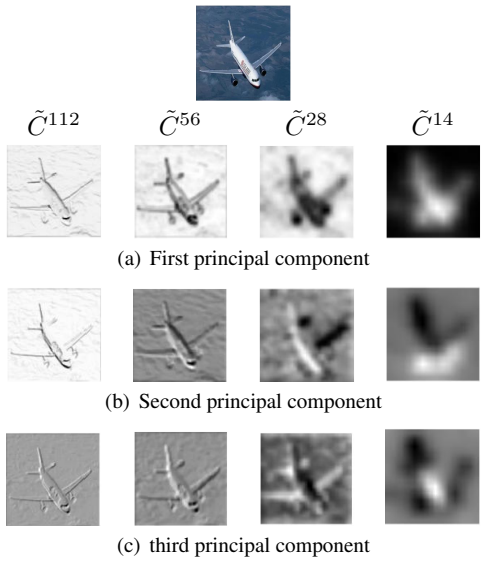
We use the shape information S_1 and S_2 to guide the upsampling process in Section 3.3. Interestingly, our numerical experiments indicate that the main shape information



(a) GoogLeNet-GAP with an input image size 224×224

(b) Upsampling, Fusion and Refinement process

Figure 2. Overview of the proposed rCAM method. The input image is fed into a GoogLeNet-GAP network, [34], operating on two different scales 224×224 and 448×448. They produce the class activation maps CAM_1 and CAM_2, the shape information maps S_1 and S_1, and the class probability maps P_1 and P_2, respectively. In the upsampling process (middle part of (b)), S_1 and S_2 are used as guidance images for a guided filter [11] that upsamples CAM_1 and CAM_2 to rCAM_1 and rCAM_2. In the fusion and refinement process (right part of (b)), rCAM_1 and rCAM_2 are combined to create rCAM_3 and finally, the rCAM is produced by applying a dense Conditional Random Field (CRF) [32] to rCAM_3.



(a) First principal component

(b) Second principal component

(c) third principal component

Figure 3. Illustration of the first three principal components of layers C^{112}, C^{56}, C^{28}, and C^{14}. While C^{112} and C^{56} yield gradient information, C^{28} and C^{14} contain mostly shape information.

can be found in the first principal component on about 70% of the images. On the other 30%, shapes can be found in the second or third principal component. Sometimes, the shapes can also be contrast inverted as shown in Figure 3. In the next section, we will show how the shapes S_1 and S_2 can be used to increase the resolution of CAMs in order to provide localization and shape information in one high resolution image.

3.3. Upsampling using guided filters

Localization results from CAMs are expressed in form of blobs of discriminative regions (see CAM_1 results in Figure 4). They may contain only parts of the objects, for example heads of the animals, rather than the whole objects' bodies. Besides that, the blob regions cannot depict the shapes well. To solve these two problems, we use the shape information recovered from GoogLeNet-GAP to guide the process of increasing the CAMs' resolution. The results we achieve are rCAMs that localize the main objects as a whole and make the objects' shapes perceivable. The increase resolution process is illustrated in Figure 2(b).

The guided filter proposed in [11] is an image processing operator that smoothens images while preserving sharp edges using a guidance image G. It relies on the assumption that inside a local window w_k that is centered at pixel x_k, there is a linear model between the guidance image G and the output image O as defined in [11]. Hence, the guided filter preserves edges from the guidance image while being independent of its exact intensity values. This is an important property because the shape information that we extract from GoogLeNet-GAP can be contrast inverted.

However, similar to non-parametric kernel regression [2], the size of the window w_k is very important. If the window size is too big, during the regression process, a large number of observations will be considered and it leads to an over-smoothed estimation of the output O. If the window size is too small, the output O will depend on too few observations and therefore, it leads to a solution with high variance. To find the optimal values for the window sizes, we estimate them on the shapes S_1 and S_2 using the variogram proposed in [7].

We assume that S_1 and S_2 follow a random process that is homogeneous and has second order stationary properties. That implies that two observations of the random process are independent of their locations and only depend on their spatial distance. To measure the spatial dependence of the

data we use the empirical variogram defined as follows:

$$\hat{\gamma}(\mathbf{h}) = \frac{1}{2|N(\mathbf{h})|} \sum_{i,j \in N(\mathbf{h})} \left(S_1(x_i) - S_1(x_j)\right)^2 \quad (1)$$

where $N(\mathbf{h})$ is the set of observations pairs (i, j) such that $\|x_i - x_j\| = h$, which is the spatial distance between two observations, and $|N(\mathbf{h})|$ is the cardinality of this set.

This empirical variogram $\hat{\gamma}$ is approximated by a model function $\gamma(h) = c_1 \cdot \left(\exp\left(-\frac{||h||^2}{2\sigma^2}\right)\right) + c_2$, which increases the generalization power of the empirical estimator. Three parameters c_1, c_2, σ are estimated such that the variogram function fits the empirical one. The σ parameter provides us information about the average size of objects. So we use σ as the size of the filter. As a result, the size of our guided filter is adapted to each image.

In order to double the resolution of a CAM using a shape prior S, we first double the size of the CAM by bilinear interpolation. Then we apply a guided filter on the upsampled CAM using S as the guidance image – a process which we denote by $G^f\left(U^2\left(CAM\right), S\right)$ where U^2 is the upscaling bilinear interpolation with a factor of 2 and G^f is the guided filter process.

We increase the resolution of CAM_1 of size 14×14 using guided filters as follows:

$$\tilde{\text{CAM}}_1^{28\times28} = G^f\left(U^2\left(\text{CAM}_1\right), S_1\right), \quad (2)$$

$$\text{rCAM}_1^{56\times56} = G^f\left(U^2\left(\tilde{\text{CAM}}_1^{28\times28}\right), S_2\right), \quad (3)$$

where S_1 and S_2 are shapes extracted from GoogLeNet-GAP and used as guidance images. The CAM_2 extracted from the higher resolution input image is of size 28×28 already and is further upsampled via

$$\text{rCAM}_2^{56\times56} = G^f\left(U^2\left(\text{CAM}_2\right), S_2\right). \quad (4)$$

As the result, we increase the resolution of both, CAM_1 and CAM_2, to $\text{rCAM}_1^{56\times56}$ and $\text{rCAM}_2^{56\times56}$ both of which are of size 56×56.

As explained in Section 3.1, CAM_1 and CAM_2 complement each other in providing coarse and fine discriminative regions – a property that is preserved during the proposed upsampling, see Figure 4. Therefore, it is beneficial to combine two of them in order to take the advantages of both.

3.4. Fusion and Refinement

Our goal is not only to provide high-resolution in localization and shape, but also to discover all the instances of the main object. In order to achieve the latter, we combine rCAM_1 and rCAM_2 which provide localization and shape information at different scales.

To do so, the $\text{rCAM}_1^{56\times56}$ and $\text{rCAM}_2^{56\times56}$ images described in the previous section are upsampled to a resolution of 224×224 pixels using bilinear interpolation. We fuse the resulting maps rCAM_1 and rCAM_2 via

$$\text{rCAM}_3 = \text{rCAM}_1 \cdot P_1(idx_1) + \text{rCAM}_2 \cdot P_2(idx_1), \quad (5)$$

where idx_1 is the index of the highest classification score of the image I_1, and P_1 and P_2 are classification probability results for image I_1 of size 224×224 and image I_2 of size 448×448, respectively. The output is rCAM_3 that combines the advantages of both rCAM_1 and rCAM_2.

Finally, to refine the accuracy of the shape prediction, we use the dense CRF implemented in [32] on rCAM_3. We first normalize rCAM_3 to $[0, 1]$ to create the probability map that indicates the presence of the main object. We use rCAM_3 and $(1 - \text{rCAM}_3)$ that represent the foreground and background probability respectively as the inputs to the CRF algorithm. The inference output from the dense CRF is our final high-resolution rCAM.

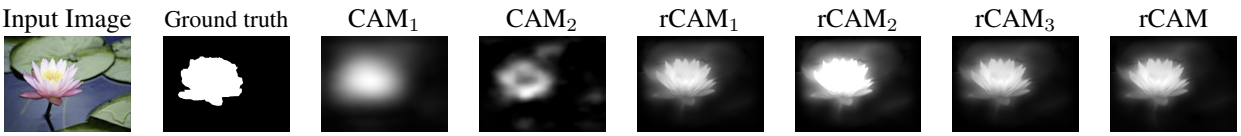
4. Evaluations

4.1. Evaluation Datasets

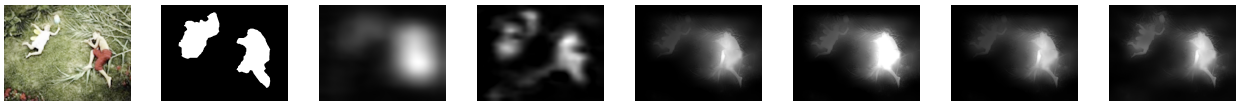
Our proposed method delivers results in two aspects: main objects' localization and shape. While many weakly-supervised learning methods output bounding boxes for objects' locations, CAM and rCAM produce probability maps (heatmaps). Therefore, instead of evaluating CAM and rCAM methods on bounding box datasets, we use three datasets: Pascal-S [19], FT [1] and ImgSal [18]. These datasets provide locations and shapes of salient objects and are commonly used to evaluate salient object detection. Each dataset has its own characteristics. While the FT dataset mainly provides a single object in each image, Pascal-S includes multiple-object images. Pascal-S is also a fair choice for the evaluation because many weakly supervised segmentation methods are trained on Pascal VOC 2012 dataset [9]. For more diversity, ImgSal contains not only single-object and multiple-object images, but also a fair amount of natural landscapes. ImgSal also contains objects that do not have the same labels as in Pascal nor ImageNet. It is the most challenging dataset for weakly supervised segmentation methods in our evaluation.

4.2. Evaluation Metrics

We use different F-measures [3] and Mean Absolute Error (MAE) [24] to analyze the performance of various CAM methods as well as weakly supervised segmentation methods. For F-measures, we use Optimal Image Scale (OIS) and Optimal Dataset Scale (ODS) [3]. OIS is computed using the best threshold for the individual image while in ODS, an optimal threshold is selected on the whole dataset. Despite the fact that OIS and ODS use different approaches in selecting optimal thresholds, both F-measures are calculated using the same formula in Eq. (6).



(a) Different CAM results on a single instance of a single object



(b) Different CAM results on multiple instances of a single main object

Figure 4. Localization and shape extraction results from various CAMs

Pascal-S								
Metric	G-Weak	CAM_1	CAM_2	$rCAM_1$	$rCAM_2$	$rCAM_3$	rCAM	% increase
OIS	0.398	0.682	0.684	0.725	0.733	0.736	**0.773**	13.34%
ODS	0.339	0.566	0.566	0.617	0.613	0.625	**0.665**	17.49%
MAE	0.395	0.298	0.338	0.290	0.314	0.291	**0.276**	-
FT								
Metric	G-Weak	CAM_1	CAM_2	$rCAM_1$	$rCAM_2$	$rCAM_3$	rCAM	% increase
OIS	0.506	0.710	0.660	0.789	0.751	0.792	**0.878**	23.66%
ODS	0.448	0.643	0.568	0.714	0.660	0.714	**0.803**	24.88%
MAE	0.367	0.223	0.280	0.206	0.250	0.215	**0.160**	-
ImgSal								
Metric	G-Weak	CAM_1	CAM_2	$rCAM_1$	$rCAM_2$	$rCAM_3$	rCAM	% increase
OIS	0.388	0.509	0.502	0.577	0.574	0.597	**0.623**	22.40%
ODS	0.273	0.419	0.417	0.491	0.478	0.505	**0.533**	27.21%
MAE	0.330	0.247	0.250	0.231	0.247	0.237	**0.188**	-

Table 1. Results of various CAM methods on Pascal-S, FT and ImgSal datasets. G-Weak [22]. CAM_1: CAM method [34] with the input size of 224×224, CAM_2: CAM method [34] with the input size of 448×448, $rCAM_1$: high resolution of CAM_1, $rCAM_2$: high resolution CAM_2, $rCAM_3$: combination of $rCAM_1$ and $rCAM_2$, rCAM: the result of applying CRF on $rCAM_3$. The best value for OIS and ODS measurements are 1. The ideal value for MAE is 0. The last column shows the relative improvement of rCAM in comparison to CAM1 for the OIS, and ODS metrics.

Dataset		Pascal-S			FT			ImgSal		
Metric		OIS	ODS	MAE	OIS	ODS	MAE	OIS	ODS	MAE
Binary Map	CCNN	0.530	0.530	0.231	0.276	0.276	0.176	0.169	0.169	**0.099**
	SEC	0.638	0.638	0.208	0.553	0.553	0.150	0.399	0.399	0.123
	TransferNet	0.735	**0.735**	**0.156**	0.714	0.714	**0.120**	0.442	0.442	0.119
Continuous Map	DCSM	0.708	0.607	0.293	0.234	0.207	0.245	0.341	0.308	0.220
	DeepLab_Box	**0.781**	0.716	0.318	0.805	0.747	0.329	0.564	0.503	0.356
	rCAM	0.773	0.665	0.276	**0.878**	**0.803**	0.160	0.623	**0.533**	0.188
	rCAM_Box	0.765	0.696	0.254	0.807	0.716	0.184	**0.663**	0.527	0.164

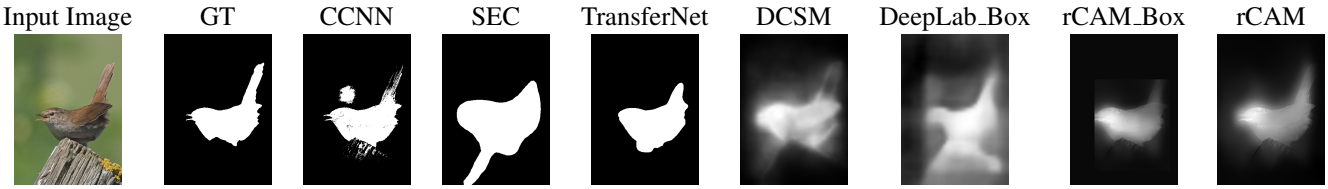
Table 2. Compasison results for different weakly supervised segmentation methods: CCNN [23], SEC [15], TransferNet [12], DCSM [27], DeepLab_Box [14] and our rCAM methods.

$$F_\beta = \frac{(1+\beta^2) Precision \times Recall}{\beta^2 \times Precision + Recall}, \qquad (6)$$

where $\beta^2 = 0.3$ as suggested in [1].

While F-measure metrics use the binarized heat map with optimal thresholds, the Mean Absolute Error (MAE) proposed in [24] measures the error of the original heat map without thresholding to the binary ground truth. The results are then averaged for all the images.

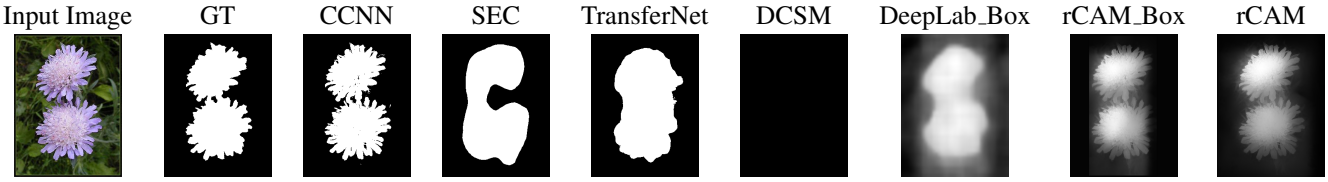
It is important to note that for F-measures, higher num-



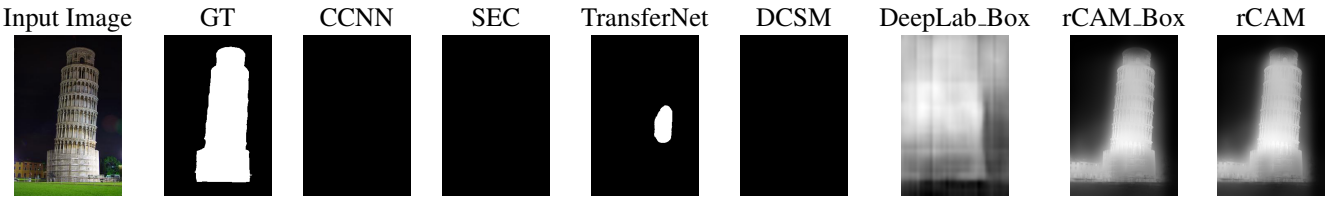
(a) Results on a single instance of a single object that is labeled in Pascal dataset

(b) Results on multiple objects that are labeled in Pascal dataset

(c) Results on multiple instances of one main object that are labeled in Pascal dataset



(d) Results on multiple instances of one main object that is not labeled in Pascal dataset



(e) Results on single object that the label is neither in Pascal or ImageNet

Figure 5. Weakly supervised segmentation results from comparison methods for various scenarios

bers indicate improved results whereas with MAE measurement, the smaller value is better.

4.3. Numerical results for various CAM methods

We analyze the performance of different CAM methods on the Pascal-S, FT and ImgSal datasets. The results in Table 1 show that the CAM method with input resolution 448×448 does not produce better results than CAM with input resolution 224×224. It can be viewed as two CAMs at two different scales complement each other, rather than compete with each other. At 224×224 resolution, CAM_1 and rCAM_1localize the main object at the largest size. At 448×448 resolution, CAM_2 and rCAM_2 can discover other smaller instances of the main object or secondary features locations of the main object, if there is only one instance (Figures 1 and 4).

However, there are significant improvements between the existing CAM methods and the high-resolution CAMs. In more details, rCAM_1 (high resolution of CAM_1) is better than CAM_1 and rCAM_2 (high resolution of CAM_2) is better than CAM_2. By combining rCAM_1 and rCAM_2, the result (rCAM_3) is better than any of the individual rCAM_1 or rCAM_2. Finally, the evaluation results are topped by applying CRF on rCAM_3 to create our final high-resolution CAM (rCAM). On the other hand, G-Weak [22] is the method that uses Global Max Pooling (GMP) [22]. The results indicate that G-Weak yields a weaker performance than the CAM method which uses Global Average Pooling (GAP), and also a weaker performance than our method.

4.4. Weakly Supervised Segmentation comparison

We divide the weakly supervised segmentation methods into two groups: the first group provides a binary segmentation for each class, the second group provides continuous values that represent the likelihood of the foreground (similar to a probability map after normalization to the range (0,1)). We call the first one Binary Map methods and the latter one Continuous Map methods. To evaluate the Binary Map methods, we set all the foreground classes to 1 and the background to 0. As the results, OIS and ODS measurements are the same for the Binary Map methods (Table 2).

The rCAM method that we describe in this paper localizes and extracts the shapes of instances of the main object at different scales. To compare with weakly supervised segmentation methods, we use Faster-RCNN [25] to retrieve bounding boxes for all detected objects. We then apply rCAM algorithm on these bounding boxes. The evaluation for this approach is called rCAM_Box.

From the numerical results in Table 2, rCAM and rCAM_Box perform better than all the competing weakly supervised segmentation methods in term of F-measures on FT and ImgSal datasets, on which none of the methods were trained. On the Pascal-S dataset, rCAM and rCAM_Box also outperform majority of the methods except DeepLab_Box [14] and TransferNet [12]. Similar to rCAM_Box, DeepLab_Box [14] method also segments object instances inside bounding boxes proposed by the Faster-RCNN network [25]. The performance of rCAM is inferior to DeepLab_Box [14] on the Pascal dataset by approximately 2-3%. It is also shown that for methods that are trained only on image labels such as CCNN [23], DCSM [27] and SEC [15], the accuracies are consistenly lower on all three datasets than the accuracies of methods that are trained using both image labels and segmentation groundtruth such as TransferNet [12] and DeepLab_Box [14]. We also observe a significant drop in performance from Pascal-S dataset to FT and futhermore to ImgSal, especially for CCNN [23], SEC [15] and DCSM [27]. This reflects the limitation of these methods to generalize beyond the datasets they have been trained on. They are prone to fail for classes they have not seen during training. The proposed method is able to maintain a much higher accuracy across different datasets without the need for any weakly supervised training or fine-tuning. It is therefore much better suited for datasets, where the training data does not need to be highly representative for the test data.

With the MAE metric, Binary Map methods such as CCNN [23], SEC [15]and TransferNet [12] have low error values. In the Continuous Map group, rCAM or rCAM_Box has the lowest errors. However, we observe that the Binary Map methods miss out more often all the segmented objects. As they cannot detect any object in an image, the output results contain only background.

Different difficulty levels of segmentation are illustrated in Figure 5. In the cases that objects' labels are in the Pascal dataset, all the methods perform relatively well even though some of the results lack some details in the shape information, e.g. CCNN [23], SEC [15] and DeepLab_Box [14] in Figure 5(b), or CCNN [23] and SEC [15] in Figure 5(c). If the object label is in ImageNet [8] but not in Pascal VOC 2012 [9] dataset, an accurate segmentation becomes significantly more challenging: In Figure 5(d), DCSM [27] is unable to detect any object, and SEC [15] as well as TransferNet [12] show degraded shape results. In the most difficult case where the object's label is neither in the Pascal VOC 2012 [9] nor in the ImageNet [8] datasets, none of the methods are able to produce reasonable results except our proposed rCAM and rCAM_Box methods (Figure 5(e)).

To understand the above results, it is important to note that all the weakly supervised segmentation methods that we use in our comparison are trained on the Pascal dataset. They all do well on Pascal but the performance drops significantly when they are evaluated on different datasets such as FT and ImgSal. Although rCAM does not need any training or fine-tuning on any dataset, its performance is already comparable to if not better than a majority of the competing methods on Pascal. Futhermore, rCAM is able to maintain the top performance on both FT and ImgSal, which demonstrates its robustness as well as the ability to generalize to a wide variety of different types of data.

5. Conclusions

In this paper, we proposed a method for extracting the localization and shape information of all instances of the main object in an image. To do so, we recover the primitive shape information from inside the GoogLeNet-GAP network. This shape information is used as guidance for the guided filter in our upsampling process to create high resolution class activation maps (rCAMs). We ascertain the benefits of using multi-scale rCAMs in our method, which does not require any extra training or fine-tuning. Our evaluation shows that, regardless of the simplicity, our proposed method outperforms existing CAM methods. Moreover, it performs on-par with competing state-of-the-art weakly supervised segmentation methods, while being far more robust to image data that is not well-represented by the training domain of the respective networks. Our experiments demonstrate that high resolution class activation maps have the potential to generalize beyond the applicability of semi supervised segmentation methods.

Acknowledgement

This research was funded by the German Research Foundation (DFG) as part of the research training group GRK 1564 Imaging New Modalities.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1597–1604, 2009.

[2] N. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):898–916, 2011.

[4] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani. Self-taught object localization with deep networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, March 2016.

[5] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly Supervised Object Localization with Multi-fold Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(1):189–203, Jan. 2017.

[7] N. Cressie. Fitting variogram models by weighted least squares. *Journal of the International Association for Mathematical Geology*, 17(5):563–586, 1985.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[10] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 646–654, 2009.

[11] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(6):1397–1409, 2013.

[12] S. Hong, J. Oh, H. Lee, and B. Han. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[14] A. Khoreva, R. Benenson, J. Hosang, and M. Hein. Simple does it: Weakly supervised instance and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *IEEE European Conference on Computer Vision (ECCV)*, 2016.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[17] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang. Weakly supervised object localization with progressive domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] J. Li, M. Levine, X. An, X. Xu, and H. He. Visual saliency based on scale-space analysis in the frequency domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(4):996–1010, 2013.

[19] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille. The secrets of salient object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 280–287, 2014.

[20] D. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999.

[21] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele. Exploiting saliency for object segmentation from image level labels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[23] D. Pathak, P. Krähenbühl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[24] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 733–740, 2012.

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[27] W. Shimoda and K. Yanai. Distinct class-specific saliency maps for weakly supervised semantic segmentation. In *IEEE European Conference on Computer Vision (ECCV)*, 2016.

[28] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich.

Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[31] 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. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6488–6496, 2017.

[32] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537, 2015.

[33] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations (ICLR)*, 2015.

[34] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.