Object extent pooling for weakly supervised single-shot localization

Gudi, Amogh; Van Rosmalen, Nicolai; Loog, Marco; Van Gemert, Jan

DOI
10.5244/c.31.36

Publication date
2017

Document Version
Final published version

Published in
British Machine Vision Conference 2017, BMVC 2017

Object Extent Pooling for Weakly Supervised Single-Shot Localization

Amogh Gudi[12]
amogh@vicarvision.nl
Nicolai van Rosmalen*[12]
nicolai@vicarvision.nl
Marco Loog[2]
m.loog@tudelft.nl
Jan van Gemert[2]
j.c.vangemert@tudelft.nl

[1] Vicarious Perception Technologies
Amsterdam, The Netherlands
[2] Delft University of Technology
Delft, The Netherlands

Abstract

In the face of scarcity in detailed training annotations, the ability to perform object localization tasks in real-time with weak-supervision is very valuable. However, the computational cost of generating and evaluating region proposals is heavy. We adapt the concept of Class Activation Maps (CAM) [28] into the very first weakly-supervised 'single-shot' detector that does not require the use of region proposals. To facilitate this, we propose a novel global pooling technique called Spatial Pyramid Averaged Max (SPAM) pooling for training this CAM-based network for object extent localisation with only weak image-level supervision. We show this global pooling layer possesses a near ideal flow of gradients for extent localization, that offers a good trade-off between the extremes of max and average pooling. Our approach only requires a single network pass and uses a fast-backprojection technique, completely omitting any region proposal steps. To the best of our knowledge, this is the first approach to do so. Due to this, we are able to perform inference in real-time at 35fps, which is an order of magnitude faster than all previous weakly supervised object localization frameworks.

1 Introduction

Weakly supervised object localization methods [3, 14] can predict a bounding box without requiring bounding boxes at train time. Consequently, such methods are less accurate than fully-supervised methods [15, 17, 18, 23]: it is acceptable to sacrifice accuracy to reduce expensive human annotation effort at *train time*. Similarly, blazing fast fully supervised single-shot object localization methods such as YOLO [23] and SSD [18] make a similar trade-off of running speed versus accuracy at *test time*. More accurate methods [15, 17] are slower and thus exclude real-time embedded applications on a camera, drone or car. In this paper we optimize for speed at train time and at test time: We propose the first weakly supervised single-shot object detector that does not need expensive bounding box annotations during train time and also achieves real-time speed at test time.

* Equal contribution as the first author.

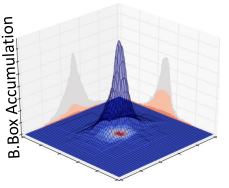


Figure 1: Accumulation of ground truth bounding boxes of Pascal VOC 2007 centered at the object's maximum activation. Note that the average extent follows a long-tailed distribution.

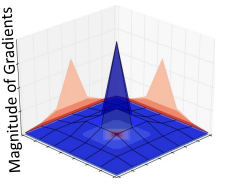


Figure 2: Gradient flow from our region pooling layer centered around the max activation. Note that our pooling follows the average extent illustrated in Figure 1.

Exciting recent work has shown that object detectors emerge automatically in a CNN trained only on global image labels [2, 20, 28]. Such methods convincingly show that a standard global max/average-pooling of convolutional layers retain spatial information that can be exploited to locate discriminative object parts. Consequently, they can predict a point inside the ground truth bounding box with high accuracy. We take inspiration from these works and train only for image classification while exploiting the spatial structure of the convolutional layers. Our work differs in that we do not aim for predicting a single point inside the bounding box, we aim to predict full extent of the object: the bounding box itself.

For predicting the object's extent, we have to decide how object parts are grouped together. Different object instances should be separated while different parts of the same object should be grouped together. Successful state-of-the-art methods on object localization have therefore incorporated a local grouping step in the form of bounding box proposals [15, 17]. After grouping, it is enough to indicate object presence and the object localization task is simplified to a bounding box classification task. In our work, we use no bounding boxes during training nor box proposals during testing. Instead, we let the CNN do the grouping directly by exploiting the pooling layer.

The pooling in a CNN groups pixels in a high-resolution image to a lower resolution one. Choices in pooling determine how the gradient is propagated back through the network. In average-pooling, the gradient is shared over all underlying pixels. In the case of a global image label, average-pooling will propagate loss gradients to all pixels in the image equally, which will cover the object but will also cover the background. In contrast, max-pooling only promotes the best point and will thus enforce only a single discriminative object part and not the object extent. Average-pooling is too wide, and max-pooling is too narrow; a regional pooling is needed for retaining the extent. Consider Fig 1, where we center the ground truth bounding boxes around its most discriminative part, given by the maximum filter response [20]. The average object extent is peaked, but has heavy tails. This motivates the need for regional pooling. In Fig 2, we show the gradient flow of our proposed pooling method centered around the maximum response. Our pooling method not only assigns gradients to the maximum or to the full image: it pools regionally.

We present the very first weakly-supervised single-shot detector. It has the following novelties. (i) **Speed**: we extend the idea of class activation maps (CAM) [28] onto a single stage CNN-only architecture for weakly supervised object localization, that achieves good accuracy while being 10-15 times faster than other related methods. (ii) **Extent pooling**: a 'regional' global pooling technique called the Spatial Pyramid Averaged Max (SPAM) pooling for capturing the object extent from weak image-level labels during training. (iii) **No region proposals**: We demonstrate a simple and fast back-projection pipeline that avoids the need for costly region proposal algorithms [26]. This allows our framework to perform real-time inference at 35fps on a GPU.

2 Related Work

Fully Supervised Object Localization. The state of the art is based on the R-CNN [9] pipeline which CNN combines the power of a classification network (e.g. ResNet [10]) with an SVM classifier and unsupervised region proposals [26]. This idea was sped up by [8] and [24] and many different algorithms emerged trying to propose the best regions [1, 7, 21], including a fully convolutional network [19] based version called R-FCN [15]. Recently published object detectors [18, 23] achieved orders of magnitude faster inference speeds with good accuracies by leaving region-proposals behind and predict bounding boxes in a single-shot. The high speed of our method is borrowed from the single-shot philosophy, albeit without requiring full supervision.

Weak Supervised Object Localization. Most methods [3, 5, 14, 27] follow a strategy where first, multiple candidate object windows are extracted using unsupervised region proposals [26], from each of which feature vector representations are calculated, based on which an image-label trained classifier selects the proper window. In contrast, our single-shot method does away with region proposals all together by directly learning the object's extent.

Li *et al*. [14] sets the state-of-the-art in this domain. They achieve this by filtering the proposed regions in a class specific way, and using MIL [6] to classify the filtered proposals. Bilen *et al*. [3] achieves similar performance by using an ensemble of two-streamed deep network setup: a region classification stream, and a detection steam that rank proposals. Wang *et al*. [27] starts with the selective search algorithm to generate region proposals, similar to R-CNN. They then use Probabilistic Latent Semantic Analysis (pLSA) [11] to cluster CNN-generated feature vectors into latent categories and create a Bag of Words (BoW) representation to classify proposed regions. The work of Cinbis *et al*. [5] uses MIL with region proposals. In our work, we also are weakly-supervised, however, we perform localization in an end-to-end trainable single-pass without using region proposals.

A recent study by [20] follows an alternate approach [16] of using global (max) pooling over convolutional activation maps for weakly supervised object localization. This was one of the first works to use this approach. Their method gives excellent result for predicting a single point that lies inside an object, while predicting its bounding boxes, via selective search region proposals, yields limited success. In our work, we focus on ascertaining the bounding box extent of the object directly. Further efforts by [2] improve upon [20] in bounding box extent localization by using a tree search algorithm over bounding boxes derived from all final layer CNN feature maps. In our work, we perform extent localization of an object by filtering CNN activations into a single feature map instead of using a search algorithm, which makes our approach faster and computationally light, achieving high-speed inference.

Finally, the concept of class activation mappings in [28] serves as a precursor to our architecture. Like us, they make the observation that different global pooling operations influence the activation maps differently. We build upon their work and introduce object extent pooling.

3 Method

To allow weak supervision training for localization for a convolutional-only neural network, we use a training framework ending in a convolutional layer with a single feature map (per object class). This is followed by a global pooling layer, which pools the activation map of the previous layer into a single scalar value, which depends on the pooling method. This output is finally connected to a two-class softmax cross-entropy loss layer (per class). This network setup is then trained to perform image classification by predicting the presence/absence of objects of the target class in the image using standard back-propagation using image-level labels. A visualization of this setup is shown in Figure 3.

During inference, the global pooling and the softmax loss layers are removed, thereby the single activation map of the added final convolutional layer becomes the output of the network, in the form of an $N \times N$ grid. Due to the flow of backpropagated gradients through the global pooling layer during training, the weights of this convolutional layer get updated such that the location and shape of the strongly activated areas in its activation map essentially have a one-to-one relation with the location and shape of the pixels occupied by positive class objects in the image. At the same time, the intensity of the activation values in this activation map essentially represent the confidence of the network about the presence of the objects at the specific location. Borrowing notation from [28], we call this single feature-map output activation a Class Activation Map (CAM).

Consequently, to extract the location of the object in the image, the CAM activations are thresholded and backprojected onto the input image to localize the positive class objects.

3.1 The Class Activation Map (CAM) Layer

The class activation map layer is essentially a simple convolutional layer, albeit with a single feature map/channel (per object class) and a kernel size of 1×1. When connected to the final convolutional layer of a CNN, the CAM layer has one separate convolutional weight for each activation map of the previous layer (see Figure 3). Training the network under weak-supervision through global pooling and softmax loss updates these kernel weight of the CAM layer through the gradients backpropagated from the global pooling layer. Eventually, the feature maps (of the previous conv layer) that produce useful activations for the training task of presence/absence classification are weighted higher, while the feature maps whose outputs are uncorrelated with the presence/absence of the positive class objects are weighted lower. Hence, the CAM output can be seen as the weighted sum combination of the activations of all the feature maps of the previous convolutional layer. Finally after training, the CAM activation essentially forms a heatmap of location likelihood of positive class objects in the input image.

The CAM layer used here is based on the concept of class activation mapping introduced in [28]. While being algorithmically similar, it should be noted that our CAM layer setup is different from the one in [28] in the following way: we perform the global pooling operation *after* the weight multiplication step (via a 1×1 conv.), while [28] does this *before* the weight multiplication step (via a FC layer). The reason for this difference is to allow greater ease of implementation and lower computational redundancy (requiring pooling on just one feature map).

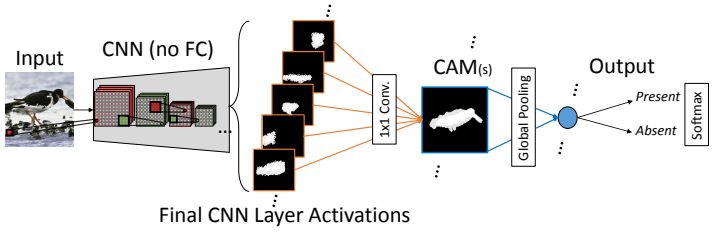


Figure 3: Visualization of the training setup for a CAM-augmented CNN. An extra conv. layer with a single feature map, the CAM, extracts the relevant feature information from the CNN's last conv layer. For weakly supervised training with present/absent annotation, the CAM is followed by a global pooling layer and connected to a softmax output/loss layer.

Algorithm 1: Fast-backprojection

Input: $[\mathbf{X}]$, $[\mathbf{Y}]$, $layer_{CAM}$, r // activation pixels in CAM layer, the CAM layer, resize ratio
Output: $bpImage$ // backprojection on input image
/* for each activation pixel in the CAM layer */
foreach $\{x,y\}$ *in* $\{[\mathbf{X}],[\mathbf{Y}]\}$ **do**
  $x_0 = x_1 \leftarrow x$; $y_0 = y_1 \leftarrow y$; $l \leftarrow layer_{CAM}$ // init
  /* loop through all layers from CAM to input */
  while $l \neq layer_{input}$ **do**
    /* s, p, k = stride, padding, kernel size */
    $\{x,y\}_0 \leftarrow \{x,y\}_0 \times s - p$
    $\{x,y\}_1 \leftarrow \{x,y\}_1 \times s - p + k - 1$
    $l \leftarrow layer_{CAM-1}$ // Go to next layer
  /* If ratio is provided, correct locations */
  if $r \neq 0$ **then**
    $\{x,y\}_0 \leftarrow \{x,y\}_0 + (\{x,y\}_1 - \{x,y\}_0) \times r/2$
    $\{x,y\}_1 \leftarrow \{x,y\}_1 + (\{x,y\}_1 - \{x,y\}_0) \times r/2$
  $bpImage[y_0 : y_1, x_0 : x_1] = 1$ // fill bpImage

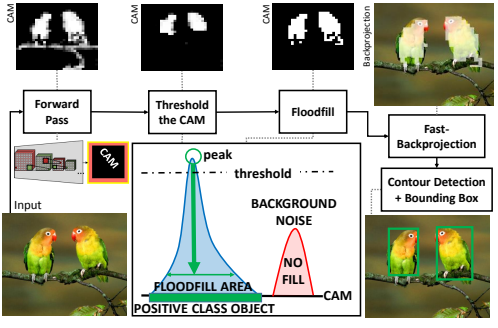


Figure 4: Visualization of the full inference pipeline. The central plot explains the thresholding and floodfilling steps. The outputs of the pipeline are positive class object bounding boxes.

3.1.1 Inference

The complete pipeline is illustrated in Figure 4. A peak of CAM's activations would occur at the location corresponding to the most discriminative part of the object. The height of the peak is related to network confidence, whereas the extent of the object is captured by the width. To get a localization proposal, we can investigate which pixels in the original image where responsible for the activations that form a peak in the CAM. First, only the CAM peaks above the CAM threshold (computed based on the ratio of biases/weights of the output layer, learnt during training) are considered. Next, using a floodfill algorithm, all activated pixels belonging to the 'mountain' of this peak (including those below the threshold) are selected, as illustrated on the central plot in Figure 4. These pixels are then backprojected onto the input image via a fast-backprojection technique explained in Algorithm 1. We call it 'fast' because it computes the mapping between CAM pixels and the input pixels without actually performing a backward pass through the network. As can be inferred, this algorithm backprojects onto all pixels in the input image that could have contributed to the CAM activations (its receptive field). Therefore, we use a ratio parameter r to influence the size of the backprojected area. This parameter can be set by heuristics, or optimised over a separate validation set. Finally, by performing a contour detection on this backprojection, we can fit simple rectangular bounding boxes on the detected contours to localize the extent of the object.

3.2 Global Pooling

During training, the gradients computed from the loss layer reach the CAM layer through the global pooling layer. The connecting weights between the CAM and the previous conv layers are updated based on the distribution/flow of the gradients defined by the type of global pooling layer used. Hence, the choice of global pooling layer and its distribution of gradients to bottom layers is an important consideration for this framework for weak supervision.

Equation Legend In the equations hereafter, we consider a CAM activation map of $N \times N$, where x_n is an arbitrary pixel in it. The backpropagated gradients from the top loss layer is denoted by g.

3.2.1 Max and Average Pooling (GMP & GAP)

Global Max Pooling (GMP) layer is essentially a simple max pooling layer commonly used in CNNs, albeit whose kernel size is the same as the input image size. During the forward pass, this essentially means it always returns a single scalar pixel whose value is equal to the pixel with the highest value in the input image. During the backward pass, Equation 1 depicts how the gradients (∇_{GMP}) are computed for all pixel locations in the CAM layer.

It can be seen from the equation that the gradient is passed only to the location with the maximum activation in the CAM. During training with a positive object image, this implies that the detectors that additively contributed in making this pixel value high are encouraged via a positive weight update. Conversely, for a negative object image, the detectors that contributed in creating the highest value in the CAM are discouraged. Therefore, the network only learns from the image area that produces max activation in the CAM, i.e., the most discriminative object parts.

$$\nabla_{GMP} = g \cdot \begin{cases} 1, & \text{if } x_n = \max\limits_{0<=n<N}(x_n) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Global Average Pooling (GAP) layer performs a similar global pooling such that the single output pixel is the average of all input pixels during the forward pass. During the backward pass, the gradients are computed as denoted in Equation 2.

$$\nabla_{GAP} = g \cdot \frac{1}{N^2} \quad (2)$$

It can be seen that every location in the CAM gets the same gradient. Due to this, over multiple epochs of training, the detectors that fire for parts of the positive class object are strongly weighted, while detectors that fire for everything else are weighted very low. Thus, the network learns from all input image locations with an equal rate due to GAP's uniform backpropagated gradient.

The visualization of the gradient flow through these pooling layers is shown in Figure 5. Due to the single-location max-only gradient distribution of the global max pooling layer, it can be hypothesised that a GMP trained CAM can be quite ideal at pointing to the discriminative parts of an object. Conversely, due to the equally spread gradient distribution of the global average pooling layer, a CAM trained with GAP would activate for the full body of object plus parts of correlated or closely situated background.

3.2.2 Spatial Pyramid Averaged Max (SPAM) Pooling

Based on the properties of the global max and average pooling layers and from a study of pooling published in [4], we propose a pooling layer that is more tuned for training a CAM network for extent localization under weak supervision.

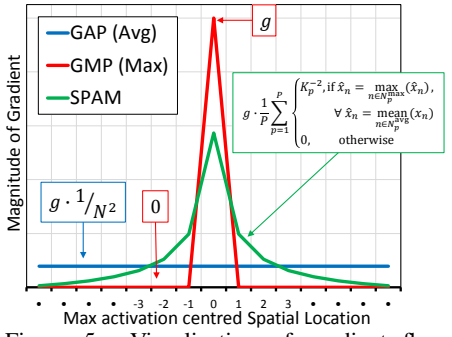


Figure 5: Visualization of gradient flow through global pooling layers. g is the backpropagated gradient from the upper later. The CAM size considered here is $N \times N$, and centered around its highest activation. SPAM pooling is considered to have P pyramid step, each with an average pooling kernel size of $K_p \times K_p$.

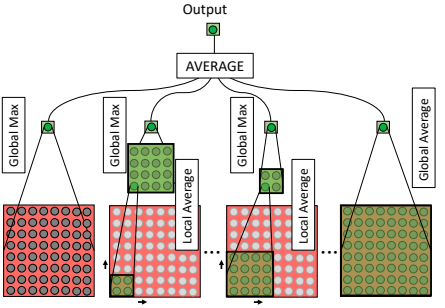


Figure 6: Architecture of the SPAM layer. First, local average pooling operations are applied in parallel with different kernel sizes, forming a pyramid of output activations. Next, global max pooling is applied and finally, its outputs are averaged. At the ends of the spatial pyramid, we directly show the equivalent GMP and GAP steps.

The approach consists of multiple local average pooling operations on the CAM activation map in parallel with varying kernel sizes. The kernel size of these average pooling operations is increased in steps (e.g., 1, 2, 4, ...), thus forming a spatial pyramid of local average pooling activation maps. Next, these activation maps are passed through global max pooling operations, which selects the maximum values among these average pooled activation maps. Finally, the output single pixel values of these combined pooling operation are averaged together to form the single scalar output of this layer. Due to the spatial pyramid structure and the use of average and max pooling operations, we call this layer global Spatial Pyramid Averaged Max Pooling, or simply SPAM pooling layer. A visualization of the architecture of SPAM layer is shown in Figure 6.

During the backward pass, the gradients are computed as depicted in Equation 3. Here, we consider a SPAM layer with P pyramid steps, each having a local average pooling kernel size of $K_p \times K_p$; the backpropagated gradients from the top loss layer is represented g.

$$\nabla_{SPAM} = g \cdot \frac{1}{P} \sum_{p=1}^{P} \begin{cases} {K_p}^{-2}, & \text{if } \hat{x}_n = \max\limits_{n \in N_p^{\max}} (\hat{x}_n), \forall \hat{x}_n = \text{mean}\limits_{n \in N_p^{\text{avg}}} (x_n) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

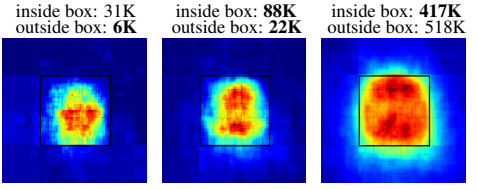
where the average/max pool kernel size at pyramid step p is $N_p^{\text{avg/max}} \times N_p^{\text{avg/max}}$.

The detectors responsible for creating maximal activation receives the strongest update, while the areas surrounding it receive an exponentially lower gradient that is inversely proportional to its distance from the maximal activation. As a result, while it strongly updates the weights of detectors of discriminative parts responsible for maximal activation, similar to GMP, it still ensures all locations receive a weak update, like in GAP. Due to this property, SPAM layer forms a good middle ground between the extremes of GMP and GAP. This can also be seen in Figure 5, which shows the gradients of SPAM layer, in comparison with that of global max and average pooling layers.

The gradient distribution of the SPAM layer is also shown in 3D in Figure 2, in comparison with the distribution of ground truth bounding boxes w.r.t the object's most discriminative part (given by CAM's maximal activation). As can be seen, SPAM's gradients are able to match the distribution of the objects' actual extent.

Method	mean Average Precision		
	Classification	Pin-pointing	Extent
GMP (Max)	**99.8**	**98.9**	69.5
GAP (Avg)	**99.4**	82.3	79.1
SPAM	**99.9**	**95.8**	**95.8**

Table 1: Results of the pooling experiments on MNIST128. Bold entries are the ones that perform 'well' on the two-class task (>95 mAP).



(a) GMP (b) SPAM (c) GAP

Figure 9: Visualization of the sum of normalized CAM activations, such that the object size present in the image is constant (denoted by the black box). The numbers denote the quantity of activated pixels (correctly) inside vs (wrongly) outside the objects' bounding box.

4 Experiments and Results

4.1 Evaluation of various Global Pooling strategies on MNIST128

Setup. As a proof of concept, we conduct experiments on a modified MNIST [13] dataset: MNIST128. this set consists of 28×28 MNIST digits placed randomly on a blank 128×128 image, thus creating a localization task. Further, we convert the 10-class MNIST classification problem to a two-class task where the digit 3 (chosen arbitrarily) is considered the positive class, and rest are negative. We consider three types of tasks: classification, bounding box localization with at least 0.5 IoU (detection/extent localization), and localization by pin-pointing. Pin-pointing is identifying any single point that falls within the object bounding box [20]. We use a FC-less version of LeNet-5 [12] with our CAM extension, trained with softmax loss via various global pooling techniques. The SPAM pooling layer used here consists of a spatial pyramid of 4 steps, with local average pool kernel sizes 1×1, 2×2, 5×5, and $N \times N$, where N is the size of the CAM activation map. After training, the layers succeeding the CAM were removed, and inference was performed as explained in 3.1.1.

The results of this experiment are in Table 1. As hypothesised, GMP is good at locating the most discriminative part of the object, and thus succeeds at pin-pointing, but fails at extent. In comparison, GAP performs worse in pin-pointing, and better in extent. The global SPAM pooling is actually able to perform fairly better overall than both the other forms of pooling for object localisation.

4.2 Experiments on PASCAL VOC

Setup We adapted an ImageNet pre-trained version of VGG-16 [25]. We replaced the fully connected layers with our CAM layer, followed by our global SPAM pooling layer plus softmax output layer. Once again, the SPAM pooling used here consisted of 4 pyramid steps with kernel sizes of 1×1, 2×2, 5×5, and $N \times N$, where N is the size of the CAM activation map. To train our CAM layer weakly on the PASCAL VOC 2007 training set, we assigned a CAM-SPAM-softmax setup, see Fig 3, to each of the 20 VOC classes. After the training, we removed the layers succeeding the CAMs, as was done in the previous experiment. We also fine-tuned the ratio parameter in Algorithm 1 on a separate validation set.

4.2.1 Analysis of CAM behaviour trained via various Global Pooling techniques

To investigate our method further, we normalize and sum the CAM activations over the whole test set (only images contained one object), such that the size of the object in all the images

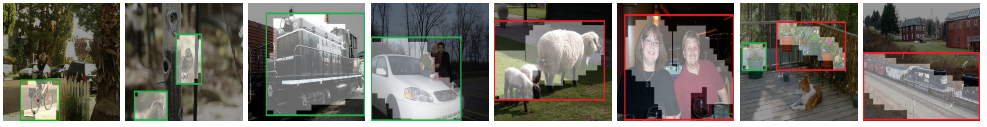


Figure 10: Localization examples: The highlighted areas in the images indicate the backprojection of CAM activations; green b.boxes match the ground truth, while red do not. Note how wrong b.box predictions are mostly either due to closely occurring objects, or closely correlated background.

Method	mAP
PASCAL VOC 2007 test set	
SPAM-CAM[Ours]	**27.5**
GMP-CAM (Max Pool)[Ours]	25.9
GAP-CAM (Avg Pool)[Ours]	15.6
Li^{RP+MIL} [14]	**39.5**
Bilen$^{RP+Ensemble}$ [3]	39.3
Wang$^{RP+pLSA}$ [27]	30.9
Cinbis^{RP+MIL} [5]	30.2
Bency$^{RP+TreeSearch}$ [2]	25.7
PASCAL VOC 2012 validation set	
SPAM-CAM[Ours]	**25.4**
GMP-CAM (Max Pool)[Ours]	22.6
GAP-CAM (Avg Pool)[Ours]	19.3
Bency$^{RP+TreeSearch}$ [2]	**26.5**
Oquab^{RP+GMP} [20]	11.7

Table 2: Detection results on PASCAL VOC 2007 & 2012. Entries marked with ^{RP} denote their use of region proposal sets.

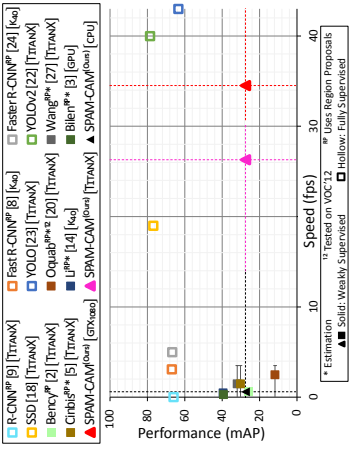


Figure 11: Speed and performance comparison between different localization methods on PASCAL VOC 2007 test set.

is constant and centered. In Figure 9, we visualize the distribution of CAM's activated pixels w.r.t the object bounding box.

Figure 9 illustrate that the GMP trained CAM activations strongly lie within the bounding region of the object, but fail to activate for the full extent of the object. Conversely, GAP trained CAM activations spread well beyond the bounds of the object. In contrast, the activations of SPAM trained CAM do not spread much beyond the object's boundaries, while still activating for most of the extent of the object. This observations support our hypothesis that SPAM pooling offers a good trade-off between the adverse properties of GMP and GAP, and hence are better suited for training CAM for weakly supervised localization.

4.2.2 Comparison with the State of the Art

The results obtained with this network can be found in Table 2, in comparison with prior work. While evaluating these results, it should be noted that all the previous work in this field rely on region proposals, which is an extra computationally heavy step. [14] uses a combination of region proposals, multiple instance learning and fine-tuned deepnets, and [3] uses region proposals and an ensemble of three deep networks to achieve this performance. In contrast, our method is purely single-shot, i.e., it requires a single forward pass of the whole image without the need of region proposals, which makes the method computationally very light. To the best of our knowledge, this is the first method to perform WSOL without region proposals.

Here, we see that the best methods [3, 14] using proposals perform significantly better. However, we are able to match the performance of other methods that also use region proposals [2, 5, 20, 27] and rely on similarly sized CNNs as ours. This observation suggests that region proposals themselves are not vital for the task of weakly supervised localization.

Speed Comparison In Figure 11, the performance of several methods is shown against the speed at which they can achieve this performance (on the PASCAL VOC 2007 test set). The test speeds for all methods have been obtained on roughly ~500×500 sized images using their default number of proposals, as reported in their respective papers. Because some studies ([5, 20, 27]) do not provide details on processing time, we make an estimation based on details of their approach (denoted by *). In the figure, we also include information on some well known fully-supervised R-CNN approaches [8, 9, 18, 22, 23, 24] for reference. As can be seen, the VGG-16 based SPAM-CAM performs about 10-15 times faster than all other weakly supervised approaches. In fact, even a CPU-only implementation of our approach roughly performs in the same speed range as other TitanX/K40 GPU based implementations. Additionally, we are able to match the speeds of existing fully supervised single-shot methods like [18, 22, 23].

5 Conclusion

In this paper, a convolutional-only single-stage architecture extension based on Class Activation Maps (CAM) is demonstrated for the task of weakly supervised object localisation in real-time without the use of region proposals. Concurrently, a novel global Spatial Pyramid Averaged Max (SPAM) pooling technique is introduced that is used for training such a CAM augmented deep network for localising objects in an image using only weak image-level (presence/absence) supervision. This SPAM pooling layer is shown to posses a suitable flow of backpropagating gradients during weakly supervised training. This forms a good middle ground between the strong single-point gradient flow of global max pooling and the equal spread gradient flow of global average pooling for ascertaining the extent of the object in the image. Due to this, the proposed approach requires only a single forward pass through the network, and utilises a fast-backprojection algorithm to provide bounding boxes for an object without any costly region proposal steps, resulting in real-time inference. The method is validated on the PASCAL VOC datasets and is shown to produce good accuracy, while being able to perform inference at 35fps, which is 10–15 times faster than all other related frameworks.

References

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 34 (11):2189–2202, 2012.

[2] Archith John Bency, Heesung Kwon, Hyungtae Lee, S Karthikeyan, and BS Manjunath. Weakly supervised localization using deep feature maps. In *European Conference on Computer Vision*, pages 714–731. Springer, 2016.

[3] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

[4] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning*, pages 111–118, 2010.

[5] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 39(1):189–203, 2017.

[6] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997.

[7] Ian Endres and Derek Hoiem. Category-independent object proposals with diverse ranking. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 36(2):222–234, 2014.

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

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

[11] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Yann LeCun et al. Generalization and network design strategies. *Connectionism in Perspective*, pages 143–155, 1989.

[14] Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang. Weakly supervised object localization with progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3512–3520, 2016.

[15] Yi Li, Kaiming He, Jian Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016.

[16] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *International Conference on Learning Representations*, 2014.

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37, 2016.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[20] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 685–694, 2015.

[21] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015.

[22] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[26] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104 (2):154–171, 2013.

[27] Chong Wang, Weiqiang Ren, Kaiqi Huang, and Tieniu Tan. Weakly supervised object localization with latent category learning. In *European Conference on Computer Vision*, pages 431–445, 2014.

[28] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.