Object Roughly There: CAM - based Weakly Supervised Object Detection

Reducing the labelling efforts for deep learned object detectors

Petra Postelnicu[1]
Supervisors: Dr. Jan van Gemert[1]**, Dr. Osman S. Kayhan**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Petra Postelnicu
Final project course: CSE3000 Research Project
Thesis committee: Dr. Jan van Gemert, Dr. Osman S. Kayhan, Dr. Petr Kellnhofer

Abstract

Highly performing object detectors require large training datasets, which entail class and bounding box annotations. To reduce the labelling effort of curating such datasets, Weakly Supervised Object Detection is concerned with training object detectors from only class labels. The most performant weakly supervised detectors (MIL-based) have high inference times, while faster methods (CAM-based) have been primarily studied in the context of localizing just one object in an image. This research proposes an extension to weakly supervised CAM-based detectors that allows them to detect multiple objects in an image and asseses their performance at localizing the full extent of objects with bounding boxes, as well as their general location with pin-points. VGG16 and a novel FPN-based classifier are experimented with as the backbone of the network, followed by GradCAM++ which indicates through heatmaps the locations of the objects predicted by the classifiers. Additionally, the proposed method is used to create pseudo-labels on which any fully supervised detector could be trained on. Results show that while the proposed method is not suitable for detecting the full extent of objects, it can accurately pin-point their general location in near real-time, thus showing the Object is Roughly There (ORT).

1. Introduction

Object detection is concerned with classifying and localizing objects in an image. As with any deep learning model, an object detector's performance highly depends on the quality of the annotations of the training dataset. However, in the context of object detection, these annotations are especially costly, since they do not only involve a class label, but also the position of the object as a bounding box. Manually annotating datasets is often the most expensive and time consuming phase of a machine learning project [19]. Moreover, simply annotating a dataset is not enough, the labels also need to be of high quality, as they determine the overall performance of the model [23]. In the case where mistakes are made, it can lead to label noise, which can affect the performance of object detectors [1].

To reduce the labelling effort, the field of Weakly-Supervised Object Detection (WSOD) [3, 28, 29] studies how object detectors can be trained without localization supervision. Unlike fully supervised object detection (FSOD) [8, 21, 38], in WSOD no bounding boxes of the objects are present in the training set, but only the classification labels. An example is presented in Fig. 1, where the FSOD model is trained with both class labels 'cat' and 'dog', as well as the bounding boxes in the image, while the WSOD model is trained only with the class labels. During inference, both models should classify and localize all the objects.

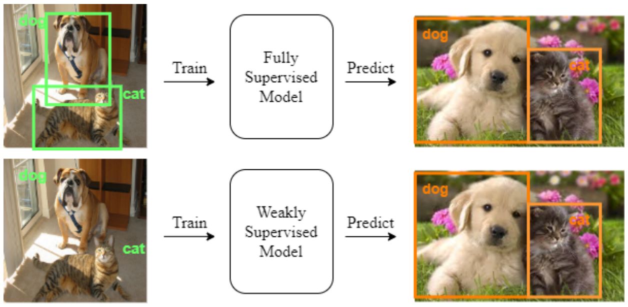

Figure 1. Difference between Fully Supervised Object Detection (FSOD) and Weakly Supervised Object Detection (WSOD).

Weakly-Supervised Object Localization (WSOL) [31, 33, 36] is a subset task of WSOD that is only concerned with detecting a single instance in an image (*i.e.* images have a single object). Shao *et al*. [25] categorize WSOD and WSOL methods into:

- Multiple Instance Learning (MIL)-based: These methods make use of a proposal generator to extract regions of an image that potentially contain an object. They treat each of these regions independently and use a CNN backbone, followed by a detection head that has a branch for classification and one for localization.
- Class Activation Mapping (CAM)-based: CAM [36] is a method that can localize where in an image a CNN-based classifier paid attention. This can be translated into heatmaps that reveal the most discriminative parts of an object, which can be transformed into bounding boxes.

Traditionally, object detection and localization entails detecting the full extent of objects with a bounding box. Pin-pointing [17] is a task that only involves detecting the approximate locations of objects. While it might not be as precise as bounding-box detection, good pin-pointing performance is still useful in applications such as tumor detection, where just the general location of the tumor can assist doctors in diagnosing patients or obstacle detection used in robotics and autonomous driving. This research treats pin-pointing as a subset of object detection.

In the context of WSOD, MIL-based methods have been extensively researched, while CAM-based methods have been primarily used for WSOL and only used for WSOD when in combination with MIL-based methods. While state-of-the-art (SOTA) MIL-based methods have gotten closer to the performance of fully supervised object detectors, they suffer from being too slow for real time object detection, which renders them unfeasible for any real world applications. Ideally, an object detector could be trained quickly with minimal supervision and have real-time inference. CAM-based methods offer the benefits of a lightweight classifier during training and can quickly evaluate an image at test time.

This research proposes a simple method that leverages CAM models for WSOD called Object Roughly There (ORT), which could be used in real world applications. A CNN-based classifier is used as a backbone of the proposed network, where VGG16 [26], as well as a novel FPN-based classifier are experimented with. A CAM-based technique called GradCAM++ [5] is applied to localize the areas in the image where the backbone classifier paid attention to. Both bounding boxes and pin points are created from these areas of interest and their precision is evaluated. Additionally, a two stage method is proposed, where the weakly supervised detector is used to generate pseudo-labels for training a fully supervised Faster-RCNN [22]. While the proposed models have low performance at detecting the entirety of objects via bounding boxes, their pin-pointing yields good results in near real-time.

As such, this research proposes a method for training object detectors in a weakly supervised manner, leveraging CAM-based techniques. More precisely the contributions of this research are as follows:

- A pipeline for performing CAM-based detection to evaluate the performance of CAM methods in WSOD, thus analyzing their potential for real-time object detection.
- A new backbone architecture for CAM-based detection, which makes use of fine and coarse-grained features.
- An evaluation of the pseudo-labeling performance of the weakly supervised method *i.e.* if it can provide candidates for a fully-supervised detector.

2. Related Research

Fully Supervised Object Detection: Object Detection is a task in Computer Vision that deals with detecting instances of visual objects of various classes in images [38]. Deep learned object detectors can be categorised into two-stage and one-stage, the core difference being that one-stage detectors can perform inference in one pass, thus being much faster. Two stage detectors use a CNN-based model to perform feature extraction, followed by region proposal generation that identifies the parts of the image that might contain an object. For each of these regions of interest, a classification head and a localization head are used to predict the class and location of objects. Faster-RCNN [22] is a SOTA two-stage detector that builds upon its predecessors, RCNN [9] and Fast-RCNN [8], being the first near real-time object detector. This research uses Faster-RCNN for the proposed two-stage method, due to its robust architecture that produces high-quality proposals. Moreover it is used as a performance upper boundary for comparison purposes.

To further improve on the lack of spatial resolution of the feature maps obtained with the CNN-based backbones of object detectors, Feature Pyramid Networks (FPN) [14] were proposed. FPNs enhance the capability of CNNs by creating a pyramid of feature maps at multiple levels, each corresponding to different scales of objects. This research uses the FPN to enhance the ability of the CAM-based models at detecting both fine and coarse-grained features.

MIL-based WSOD: A main challenge in WSOD is the Multiple Instance problem, which involves detecting multiple objects of the same class in a single image. MIL-based methods tackle this problem by treating an image like a bag of region proposals. If the image is labeled as positive for a certain class, at least one of the region proposals must contain an object of that class; if not, none of them do. During training the highest scoring proposals that came from a positive image are used as pseudo-positive examples to update the classifier, while the proposals that came from negative images are used as negative examples. The loss function aims to increase the scores for the selected positive regions and decrease the scores for the negative ones. WSDDN [3] is a popular MIL detector, which leverages a CNN-based network to perform feature extraction and a proposal generator [30, 37] to extract region proposals. Each region proposal is processed independently with separate classification and detection data streams. The classification stream is responsible for computing the probability of each class being in the region proposal and the detection stream predicts every proposal's existing probability score for each class. To further improve on the performance of WSDDN, PCL [29] generates proposal clusters which aid the model in detecting the full extent of objects, not only the most discriminative parts. PCL is used in this research to provide a comparison to the proposed CAM-based models, which are designed to be much faster.

A popular technique to improve the performance of MIL-based detectors is transforming WSOD to FSOD, which involves using a weakly supervised detector to generate pseudo ground-truth bounding boxes on which a fully supervised detector can be trained. Several strategies exist for mining the best proposals for the pseudo labels: selecting the boxes with the highest score, selecting boxes with the maximal relative improvement score between two epochs [11], or merging together small boxes [34]. The WSOD to FSOD technique is used in the proposed two stage method, however, no pseudo-ground-truth mining is needed as the proposed ORT generates few bounding boxes.

CAM-based WSOL: Class Activation Maps (CAM) [36] is a method that can localize objects in an image, under the observation made by Zhou *et al.* [35] that CNNs encode the location of objects despite having no localization supervision. This provides a base for all CAM-based WSOD methods. First a CNN-based classifier is modified (VGGnet [26], AlexNet [13] and GoogLeNet [27] have been experimented with) by taking out the fully connected (FC) layers and replacing them with a Global Average Pooling (GAP) layer followed by a FC softmax layer. While these modifications decrease the classification performance of the networks, they are necessary in order to compute the class activation maps. To obtain the CAM for a class, each feature map from the last convolutional layer is multiplied by its weight and then summed together. A ReLU operation is then applied to filter out negative activations. The class activation maps are represented as heatmaps which highlight the most discriminative parts of an image that the classifier focused on. By thresholding the heatmap at a certain procentage of its maximum, the image is segmented into the part that contains the object, which determines the bounding box. While this CAM method is very fast, it had two main problems: the need to modify the structure of the network, as well as the discriminative region problem, which means that the detector primarily focuses on the most discriminative regions of the objects.

Several methods that change the way class activation maps are computed to improve on these problems have been proposed. Selvaraju *et al*. propose GradCAM [24], which uses different feature map weights to calculate the class activation maps. While CAM directly uses the weights from the fully connected layer, GradCAM uses the gradients of the output class score with respect to the feature maps to compute the importance weights. However, GradCAM has problems with highlighting objects completely when multiple instances of that object appear in the same image. GradCAM++ [5] was proposed in response to this problem, by introducing gradient outputs that are weighted for pixels in specific locations. GradCAM++ computes the weights for each pixel using second-order derivatives, which helps in assigning more precise importance to different regions of the feature maps. This research uses GradCAM++ as a means to compute the class activation maps, because of its speed and increased capability at covering the full extent of the object and at detecting multiple instances.

Wei *et al*. recognised one of the causes of the discriminative region problem of CAM is that it only considers the features from the last convolution, not takeing advantage of the shallow features from the previous layer. They proposed Shallow feature-aware Pseudo supervised Object Localization (SPOL) [31], which aggregates the CAMs obtained from shallow and deep layers to filter out background noise and generate sharped object boundaries. LayerCam [10] was also proposed as a method that can take into consideration features from both shallow and deep layers, thus gathering information ranging from coarse-grained localization to fine-grained details. In this research, the architecture of the classifier itself is used to aggregate the features from shallow and deep layers, by leveraging FPNs [14].

Other CAM-based methods optimize the post processing of the class activation maps. Rethinking CAM [2] introduces Percentile as a Standard for Thresholding (PaS), which thresholds the heatmap using a procentage of a procentaile of the CAM, instead of the maximum value. Kim *et al*. propose Inferior Value Removal (IVR) [12] as a normalization method for the CAM heatmaps, which builds upon PaS. IVR adjusts activation map values to maintain consistent importance of regions, despite variations in the original map's maximum and minimum values. The proposed method also uses IVR to normalize the images, as it is a cost-free optimization.

Weakly Supervised Pin Pointing: Pin pointing means indicating where the object might be with a point instead of a bounding box. It was introduced by Oquab *et al*. [17] when they proposed a WSOD method similar in concept to MIL-based methods. However, because their detector was not trained to detect the full extent of objects, they quantified the detection performance using pin-pointing. This is evaluated the same way as bounding boxes, with the exception that a point is considered to be correct if it lies within the ground truth bounding box. Note that in their method, they did not detect multiple instances of the same object in an image. This research also evaluates pin-pointing capabilities for the proposed CAM-based models.

3. Method

3.1. One-stage Detection

The general inference pipeline of the proposed model ORT can be observed in Fig. 2. Suppose we have dataset $\mathcal{I}$ of N training images in C classes. The set is given as $\mathcal{I} = \{(\mathbf{I^1}, \mathbf{y^1}), \ldots, (\mathbf{I^N}, \mathbf{y^N})\}$ where $\mathbf{I^k} \in \mathbb{R}^{H \times W \times 3}$ is an image with height H, width W and 3 RGB channels and $\mathbf{y}^k = [y_1, \ldots, y_C] \in \{0, 1\}^C$ is a vector of labels indicating the presence or absence of each class in image $\mathbf{I^k}$. First a classifier is used to infer the class labels $\mathbf{y^k}$ present in an image $\mathbf{I^k}$. Then, GradCAM++ [5] is used to detect the location of objects of each predicted class where the class probability is over a threshold t. This threshold is used to only detect the location of the objects that are most probable to be in the image. The obtained heatmap is segmented to retrieve the parts of the image that were most significant during classification. From these areas of interest, bounding boxes are constructed.

Classifier Architecture: A classifier with a CNN backbone, followed by a Global Average Pooling (GAP) layer and a classification head is trained to detect what object classes $\mathbf{y^k}$ are present in an image $\mathbf{I^k}$. As a baseline, an unmodified VGG16 [26] is used, as previously done in [5, 10, 24]. Inspired by the architecture of Faster-RCNN [22] this research also proposes a classifier that is able to aggregate the features from both shallow and deep layers, by using an FPN [14] with a ResNet-50 backbone.

To compute the aggregated feature maps, the FPN makes use of two pathways: bottom-up and top-down, as can be seen in Fig. 3. The bottom-up pathway is a standard forward

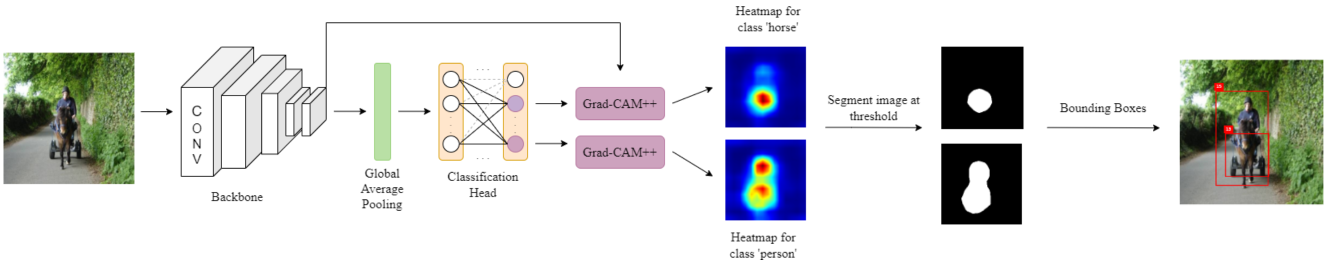

Figure 2. Proposed one-stage detection pipeline: A classifier made up of a CNN backbone, followed by a Global Average Pooling (GAP) layer and a classification head with multiple Fully Connected (FC) layers is used to extract feature maps from an image. GradCAM++ backpropagates the predicted class scores to the final convolutional layer for each class to obtain the Class Activation Maps (CAMs) represented as heatmaps. The heatmaps are segmented and contour detection is used to extract the locations of the objects.

pass through the ResNet-50 backbone, which is composed of 5 convolutional modules, each containing many convolutional layers. Let the output of each of these modules be denoted as $C1, C2, C3, C4, C5$. As we go up in the bottom-up pathway, the spatial resolution decreases, while the semantic value increases. The top-down pathway starts by upsampling the most semantically rich, but spatially coarser feature map $C5$. As we go down, lateral connections from corresponding bottom-up feature maps are added to these upsampled maps. Specifically, $C5$ is upsampled and combined with $C4$ to form $P4$. This combined map is then upsampled and added to $C3$ to form $P3$, and $P2$ is formed by combining the upsampled $P3$ with $C2$. Note that the top-down pathway stops at $P2$, as $C1$ is too large and would slow the process down too much. Each of the combined feature maps $P2, P3, P4, P5$ captures both high-resolution spatial information and strong semantic content.

To create the classifier, one of the feature maps resulted from the FPN is followed by a convolution, which becomes the target layer for GradCAM++. A GAP layer and the classification head of VGG16 are added to the network, as presented in Fig. 3. Each of the feature maps $P2, P3, P4, P5$ are used as part of different backbones of ORT.

Multi-label multi-class loss: When dealing with object localization tasks *i.e.* one image contains only one object, Cross-Entropy Loss is most commonly used. However, for the object detection task where multiple objects of different classes can be present in the same image, the loss also needs to be multi-label on top of being multi-class. This means that the classes in an image are not mutually exclusive. To account for this, the proposed model uses a Binary Cross-Entropy (BCE) Loss, which is a measure of the difference between the true labels and the predicted labels for each class in each image. It is computed as the as the sum of C binary logistic regression loss functions, where a class probability is obtained by applying the sigmoid function to the logits predicted by the classifier. For one training sample I^k this is more formally defined as:

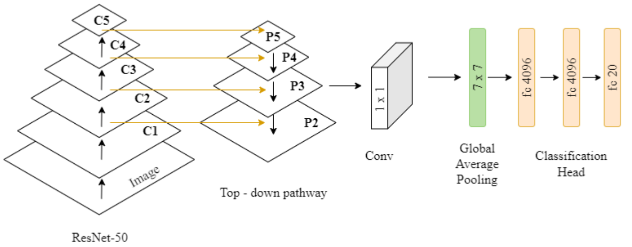

Figure 3. Proposed architecture for FPN Classifier: The ResNet50 backbone serves as a bottom-up pathway encoding the image into feature maps across five modules. The top down pathway upsamples the resulted low resolution feature maps and aggregates them with the corresponding bottom - up pathway maps via lateral connections. One of the resulting feature maps containing spatial and semantic information is passed through a 1 x 1 convolution, followed by a 7 x 7 GAP layer and 3 FC layers used for classification.

$$\mathcal{L} = -\sum_{c=1}^{C} y_c^k \cdot \log \sigma(x_c^k) + (1 - y_c^k) \cdot \log(1 - \sigma(x_c^k))$$
$$\sigma(x) = \frac{1}{1 + \exp(-x)}, \tag{1}$$

where x_c^k is the predicted logit for class c in image I^k and $\sigma(x_c^k)$ is the sigmoid function applied to the logit x_c^k.

Obtaining Class Activation Maps: To compute the class activation maps, GradCAM++ [5] uses the predicted class score x_c^k, before it is passed through the sigmoid function. It then backpropagates this score to the last convolutional layer of the classifier, which is called the *target layer*. The calculations are based on the assumption that the predicted class score, denoted as x^c in the following equations, can be written as a linear combination of its global average pooled last convolutional layer feature maps A_f:

$$x^c = \sum_f w_f^c \cdot \sum_{i=1}^{H} \sum_{j=1}^{W} A_f(i,j), \tag{2}$$

where $A_f(i,j)$ is the activation of the f-th node in the target layer at the spatial location (i,j).

As such, each spatial location (i,j) in the class specific heatmap L^c is calculated as a weighted sum of all feature maps $A_f(i,j)$ in the target layer:

$$L^c_{\text{Grad-CAM++}}(i,j) = \sum_f w^c_f A_f(i,j), \tag{3}$$

where w^c_f is the weight coefficient that represents the importance of the f-th feature map A_f for class c. This is defined as:

$$w^c_f = \sum_{i=1}^{H} \sum_{j=1}^{W} \alpha^c_f(i,j) \cdot \text{ReLU}\left(\frac{\partial x^c}{\partial A_f(i,j)}\right), \tag{4}$$

where $\alpha^c_k(i,j)$ is a scaling factor that adjusts the importance of each activation based on higher-order derivatives of the class score x_c with respect to the activations $A_f(i,j)$.

Inferring Object Location: The first step in transforming the heatmaps obtained with GradCAM++ [5] is normalization. For this, IVR [12] normalization is used, which divides all values in the class specific heatmap L^c by the maximum value, after a percentile value from its minimum value is subtracted. This can be expressed as follows:

$$\mathbf{L^c}' = \frac{\mathbf{L^c} - \text{Pct}_p(\mathbf{L^c})}{\max(\mathbf{L^c} - \text{Pct}_p(\mathbf{L^c}))}, \tag{5}$$

where p represents the procentile up to which values are excluded from the original heatmap L^c.

To segment the most significant regions in the normalized heatmap, a binary threshold is applied. This threshold ensures that only areas exceeding a percentage τ of the maximum value in the normalized heatmap are retained.

Contour detection is applied on the segmented heatmap and bounding boxes are placed around the identified contours. To avoid very small boxes being generated as a result of noise, the contours that cover under 1% of the whole image area are excluded. To ensure tight bounding boxes around the objects, if the contours obtained are bigger than 80% of the image area, the image is recursively segmented with an increased τ for a maximum of 5 iterations.

When creating pin-points, the most natural approach would be placing the point at the maximum value of the heatmap. However, doing so would eliminate any chance of detecting multiple instances of the same class. Instead, a high segmentation threshold τ' is used, such that only the most significant pixels in the image remain. These represent the most discriminative areas detected in the image. Contour detection is then applied, and the pin points are placed at the highest value within the contour.

Confidence Score: Fully supervised object detectors have a confidence score corresponding to each prediction. This is calculated per bounding box as the class probability. However, since ORT doesn't make class predictions on regions of interest, the class probability in the whole image is used instead. This means that within an image, each detected object of the same class will have the same score.

3.2. Two-stage Detection

Following the WSOD to FSOD framework a two stage model is proposed for CAM-based methods, which can be seen in Fig. 4. First, pseudo-labels for detection are generated for the train set using the same method as the one stage detection. Having these pseudo-labels, any object detector can be used in the second stage. In this research the performance of FasterRCNN [22] is analyzed.

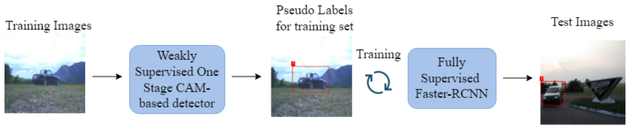
Training Images
Weakly Supervised One Stage CAM-based detector
Pseudo Labels for training set
Training
Fully Supervised Faster-RCNN
Test Images

Figure 4. Proposed two stage detection pipeline: The one stage weakly supervised object detector is used to generate pseudo bounding boxes for the training set. A fully supervised Faster-RCNN is then trained and used to make predictions.

4. Experiments

4.1. Experimental setup

Dataset: All experiments for the proposed method are done on PASCAL VOC 2007 [6], which is a large scale dataset for objects. It contains 5011 trainval images and 4092 test images with 20 classes. For an in depth investigation of the strengths and weaknesses of the proposed models, evaluation is also conducted on certain splits of the dataset. More precisely, one split for localization (1905 samples) is created from the images that only contain one object, which is further broken down into small (507 samples) and large (1398 samples) objects. Objects with a ground-truth bounding box area under 20% of the whole image area are considered small, while the rest are considered large. A split is created for multiple instance (936 samples) images, which contain multiple objects of the same class. The remaining images make up the multi-class (2111 samples) data split *i.e.* images with multiple objects of multiple classes.

Metrics: To measure the classification performance, Multi-label Average Precision is used. For the localization with bounding boxes, that standard object detection Mean Average Precision (mAP) is computed. When computing the average precision, a predicted bounding box is considered correct if it has Intersection over Union (IoU) of more than 50% with ground-truth box (mAP@50). When measuring the pin-pointing performance, mAP is modified such that a point is considered correct if it is inside the ground truth bounding box. Correct Localization (CorLoc) is used to evaluate the localization accuracy of the weakly supervised detectors on the trainval dataset. It computes the precision of the predicted bounding boxes, where a detection is considered correct if it has IoU of more than 50% with the ground truth bounding box corresponding to the same class.

Training procedure: Because VOC 2007 is a relatively small dataset, the VGG16 and the ResNet50 backbone of the FPN classifier are pretrained on the ImageNet dataset. Faster-RCNN is also pretrained on the COCO MS dataset, in order to facilitate a fair comparison. The VGG16 and FPN classifier are then fine-tuned on the VOC 2007 dataset. SGD with momentum of 0.9 and weight decay of 0.0005 is used as an optimizer. The network is trained with a batch of 16 samples over 20 training epochs with a learning rate starting from 10^{-4}. The learning rate decreases by 90% when the validation loss stops improving. The Faster-RCNN is trained in the same fashion, but with a learning rate starting from 0.005. Trivial Augment Wide [15] data augmentation technique was used to increase the models' generalizability and ability to learn more robust features.

Inference hyperparameters: GradCAM++ is used only for the classes with a probability over the threshold $t = 0.5$. Similarly, the Faster RCNN predictions considered have a confidence score ≥ 0.5. When normalizing the CAM heatmap, the IVR procentile is set at $p = 0.3$. When segmenting the heatmap to obtain bounding boxes and pin points, the thresholds $\tau = 0.2$ and $\tau' = 0.5$ are used.

4.2. Classification Results

As the performance of the proposed weakly supervised method highly depends on the performance of the classifier used, the Multilabel Average Precision of the different classifiers is evaluated first. From the results presented in Tab. 1, it can be observed that the different classifiers have similar performance (between 83.5% and 86.9%), with the FPN using the P4 feature map having the best overall score.

Method	Full dataset	Localization			Multi Instance	Multi Class
		small	large	both		
VGG16	83.5	71.2	86.6	85.0	81.6	78.3
FPN P5	85.5	77.5	89.1	87.7	83.7	81.6
FPN P4	86.9	76.3	90.2	88.1	84.2	83.5
FPN P3	86.0	77.8	89.9	88.2	82.7	81.9
FPN P2	86.3	76.9	89.7	88.1	83.3	82.6

Table 1. Classification Multilabel Average Precision on VOC 2007 of the VGG16 and the FPN based classifier when using the 4 different feature maps. The classifiers used in the proposed models manage to effectively identify the objects present in an image, with FPN P4 having the best performance.

4.3. Detection Results

The proposed weakly supervised detection models are evaluated with the standard mAP@50 metric on the VOC 2007 dataset. Table 2 shows the bounding box detection performance of the proposed models with the one-stage and two-stage methods compared to the fully supervised Faster-RCNN and a SOTA weakly supervised MIL model PCL [29], while Tab. 3 shows the detection performance of the prroposed models on the trainval set. The proposed models underperform compared to Faster-RCNN and PCL. However, the FPN-based models perform significantly better overall than ORT-VGG16, by up to 6.4% overall, showing that a ResNet50 backbone is more suitable at extracting features for this task, with the exception of small localization objects. When considering the 4 different feature maps from the FPN, it seems the model doesn't benefit from incorporating features from deeper layers. Note that the segmentation threshold used at inference time was the same across all models and a more suitable threshold should be used for each separate FPN feature map in order to yield better results when incorporating the deeper layers. When comparing the one stage and two stage results, it can be seen that the two stage method brings a significant performance increase, between 8.4% and 15.1% across the whole test set.

Method	Full dataset	Localization			Multi Instance	Multi Class
		small	large	both		
ORT-VGG16	6.0	27.2	10.9	12.2	4.2	5.5
+Faster-RCNN	21.1	49.4	28.5	32.6	15.3	21.4
ORT-FPN P5	12.4	21.5	39.8	33.0	6.2	11.4
+Faster-RCNN	20.8	23.2	54.9	43.1	12.0	19.7
ORT-FPN P4	10.4	14.2	33.6	25.0	4.9	10.3
+Faster-RCNN	22.2	27.3	53.2	43.4	14.5	20.1
ORT-FPN P3	6.7	21.2	17.6	15.8	3.7	6.9
+Faster-RCNN	20.5	38.1	38.2	36.5	14.3	20.3
ORT-FPN P2	9.2	19.2	25.3	20.7	6.0	8.6
+Faster-RCNN	22.0	33.7	49.4	43.8	14.8	20.3
PCL*	48.8	-	-	-	-	-
Faster-RCNN	74.2	87.4	82.5	85.1	67.4	71.0

Table 2. Object Detection results with mAP@50 on VOC 2007. From top to bottom: the proposed weakly supervised models using the VGG16 classifier and the FPN classifier with different feature maps, where '+' indicates the two stage method. At the bottom, the SOTA weakly supervised MIL method PCL, where * indicates the result is taken from the original paper and the fully supervised Faster-RCNN. Weakly supervised models are outperformed by the fully supervised detector, struggling most with multi instance and multi class images. The two stage method boosts the performance of the one stage models.

Analyzing the qualitative results presented in Fig. 5, we can see that the VGG16 model focuses more on the

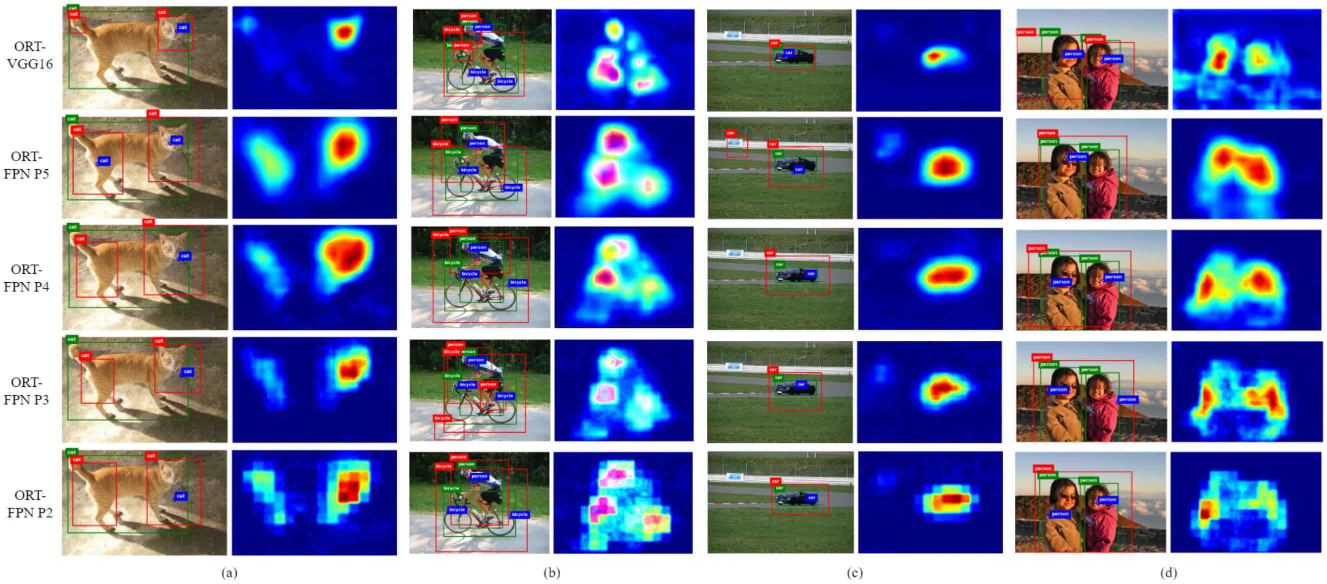

Figure 5. Comparison between the detection and pin pointing performance between the different backbone classifiers used for ORT on a a) large localization image, b) multi instance image, c) small localization image and d) multi class image of the VOC 2007 test set. Each column contains the original image with the ground truth bounding boxes in green, the predicted bounding boxes in red, the predicted pin points in blue and the heatmap generated with GradCAM++. Note that the heatmaps for multi class images are shown by overlaying the heatmaps from the different classes for visualization purposes. The FPN-based models manage to detect more of the objects, compared to the VGG16 that mainly looks at the most discriminative parts. The deeper feature maps in the FPN detect more fine-grained features and have a less uniform aspect.

Method	Full dataset	Localization			Multi Instance	Multi Class
		small	large	both		
ORT-VGG16	17.8	34.4	11.8	15.1	18.0	18.8
ORT-FPN P5	26.4	22.4	35.6	31.2	20.0	24.3
ORT-FPN P4	23.6	23.1	32.6	28.5	17.7	21.7
ORT-FPN P3	18.9	26.4	18.4	18.8	19.3	18.6
ORT-FPN P2	20.8	24.0	24.0	22.4	16.7	20.3

Table 3. CorLoc results on the VOC 2007 trainval set of the proposed one stage method with the VGG16 and the FPN based classifier when using the 4 different feature maps. The proposed models struggle at detecting the full extent of objects. The FPN backbone performs overall better than the VGG16 one.

most discriminative parts of the objects. This allows for tighter bounding boxes around smaller objects, which explains why it performs better at small localization. However, this decreases the performance at large localization, where the model needs to be able to see the full extent of the object. When looking at multi class detection, all the models seem to incorporate features from other classes when looking at the target class. Note that this is due to the used BCE loss, which doesn't explicitly encourage the model to separate the classes from one another. While considering the multi instance images, the models struggle with separating the individual objects, especially when they are close together. Looking at the heatmaps obtained from the 4 different FPN feature maps, they become more fine-grained and less uniform as deeper layers get incorporated. While heatmaps obtained with P2 tend to be very noisy, with P3 the shape of the objects seems to be defined best.

4.4. Pin Pointing Results

While the proposed models have low performance at detecting the full extent of objects, when tasked with just providing their general location in the form of pin points they show promising results. To create pin points from the predictions of the Faster-RCNN, the center of each of the predicted bounding boxes is considered. As can be seen in Tab. 4 the performance gap between fully supervised and weakly supervised models significantly decreases. As opposed to bounding box detection which is highly dependent on the segmentation threshold, pin pointing benefits from incorporating features from deeper layers. The P4 feature map of the FPN has the best overall performance at 85.6% on the whole test set, followed closely by P2. The VGG16 still yields top performance at localizing small objects, while the first 3 FPN feature maps perform the best at localizing large objects with up to 99.3% mAP, which is marginally better than the Faster RCNN. Looking at the qualitative results in Fig. 5, pin pointing benefits from detecting the most discriminative parts of objects. When it comes to multi instance images, the more aggressive threshold manages to better separate the instances. Further detection experiments performed on images resized to 224x224, as well as a more in-depth qualitative analysis are present in the Supplement.

Method	Full dataset	Localization			Multi Instance	Multi Class
		small	large	both		
ORT-VGG16	80.0	75.6	98.8	96.5	56.1	71.3
+Faster-RCNN	92.5	83.6	99.4	98.6	79.4	87.0
ORT-FPN P5	79.7	73.8	99.3	96.9	53.7	72.1
+Faster-RCNN	88.2	88.1	99.0	98.3	71.2	81.9
ORT-FPN P4	85.6	75.3	99.0	96.6	61.0	78.1
+Faster-RCNN	89.7	89.5	99.1	98.1	73.8	84.3
ORT-FPN P3	83.8	72.7	99.3	96.9	59.7	75.3
+Faster-RCNN	92.6	85.1	99.0	97.9	77.8	87.2
ORT-FPN P2	85.0	71.5	98.6	96.5	61.4	77.0
+Faster-RCNN	91.2	87.5	99.1	97.6	75.8	85.8
Faster-RCNN	95.7	90.0	99.0	99.3	85.1	92.2

Table 4. Pin Pointing results with mAP on VOC 2007, where a point is considered correct if it falls in the ground-truth bounding box. From top to bottom: the proposed weakly supervised models using the VGG16 classifier and the FPN classifier with different feature maps, where '+' indicates the two stage method. At the bottom, the fully supervised Faster-RCNN. The proposed models can successfully pin point the general location of objects, without knowing the objects' locations during training, their performance being close to the fully supervised detector.

4.5. Real-time Inference Analysis

The inference speed of ORT is evaluated to analyze its potential for real time detection. In Tab. 5 the results are compared to SOTA MIL-based weakly supervised detector PCL [29], as well as to fully supervised object detectors Faster-RCNN [22] and YOLOv3 [20]. Note that YOLOv3 can perform real time object detection. The inference time in seconds is averaged across 100 random images from the VOC 2007 test set, on a Intel(R) Core(TM) i7-10750H CPU.

Method	ORT-VGG16	ORT-FPN P5	PCL	YOLOv3	Faster-RCNN
Inference time (seconds)	2.38	1.31	36.35	0.59	1.81

Table 5. Comparison between the inference time (in seconds) of the proposed weakly supervised models, SOTA weakly supervised MIL - based model (PCL) and two stage (Faster RCNN) and one stage (YOLOv3) fully supervised detectors.

While the library code for Faster-RCNN and YOLOv3 is highly optimized, the ORT is not and thus its inference time has potential to be even faster. Note that the speed of the proposed models is highly dependent on the accuracy of the backbone as GradCAM++ and the bounding box generation are used for as many classes as predicted, which explains why the FPN-based model is faster than the VGG16 one. The results in Tab. 5 show that the proposed models can be considered near real-time, being by far faster than MIL-based methods. Additionally, the FPN-based models achieve lower inference time than Faster-RCNN.

5. Discussion

This paper introduced a pipeline that extends CAM-based methods to perform weakly supervised object detection. The method provides flexibility in both the choice of backbone classifier, as well as the choice of method to compute class activation maps. Experiments were performed with the VGG16 classifier, as well as a novel FPN-based classifier that incorporates both fine-grained and coarse-grained features of an image. Results showed that the ORT has reduced capabilities at detecting the full extent of objects with bounding boxes, but it can achieve good pin-pointing performance: 85.6% and 92.6% mAP@50 on the VOC 2007 dataset with the one- and two-stage method respectively. Moreover, its near real-time inference speed shows potential for real world applications in fields such as robotics and autonomous driving.

The inference hyperparameters, such as the segmentation thresholds, highly affect the performance of ORT and would benefit from being tuned to the different classifier backbones, especially across the FPN feature maps. While the two stage method does increase the performance of the detectors, the influence of the data used to pretrain the fully supervised detector should be analyzed. The choice of algorithm used to create the CAMs, aside from GradCAM++, should also be experimented with. Better training strategies for weakly supervised CAM-based detectors, such as *Easy-to-hard*, as used by Zhang *et al*. [32], could improve the performance by gradually increasing the difficulty of the samples during training. Moreover, a loss function that better separates the classes could be used, such that the classifier can learn to better focus on individual classes and not take into consideration features from others.

Future research should improve bounding box generation by using strategies from fully supervised object detection, such as Non-Maximum Suppression (NMS) [16], which merges predicted bounding boxes based on confidence scores. The confidence scores of ORT would have to also account for the location of the objects, such that detections of the same class within the same image get different confidence. Finally, the proposed method should be analyzed when using a transformer architecture instead of the simple classifier. In object detection, transformer based models such as DETR [4] and DINOv2 [18] have reached state-of-the-art, even in self-supervised settings.

6. Responsible Research

Reproducibility: To ensure reproducibility when it comes to the data used, PASCAL VOC 2007 [6] adheres to the principles of *FAIR*. It is *findable* through its dedicated webpage[1] and *accessible* as it is available for download without any restrictive bariers, both from the website as well as through its PyTorch library[2].Moreover, the dataset is *interoperable* because it is formatted in a standard, widely-used format (XML for annotations and JPEG for images), which allows for seamless integration with various machine learning frameworks and tools. Lastly, it is *reusable* due to its comprehensive documentation[3].

The codebase used for this research is publicly available on GitHub[4], along with the experiments' logs. To ensure reproducibility, the code is commented and comes with comprehensive instructions in the *README*. Random data shuffling and sampling was controlled with the use of a fixed generator seed of 42, meaning that all data splits used can be easily reproduced.

Integrity: This research adheres to the 5 research integrity principles outlined in the Netherlands Code of Conduct [5]: *honesty, scrupulousness, transparency, independence* and *responsibility*. All aspects of this work were presented in a truthful manner, acknowledging the limitations of the method. The research was thoroughly conducted, assuring transparency in the methodology. Furthermore, there were no external influences, as well as no ethical concerns associated with this research. The pre-existing code that was used is all publicly available, either through the PyTorch library (VGG16, FPN, Faster-RCNN) or within public GitHub repositories (GradCAM++ [7], PCL [6]) which are licensed under MIT.

References

[1] Bishwo Adhikari, Jukka Peltomäki, Saeed Bakhshi Germi, Esa Rahtu, and Heikki Huttunen. Effect of label noise on robustness of deep neural network object detectors. In *International Conference on Computer Safety, Reliability, and Security*, pages 239–250. Springer, 2021. 1

[2] Wonho Bae, Junhyug Noh, and Gunhee Kim. Rethinking class activation mapping for weakly supervised object localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 618–634. Springer, 2020. 3

[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. 1, 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 8

[5] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 2, 3, 4, 5

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 5, 9

[7] Jacob Gildenblat and contributors. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021. 9

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 2

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015. 2

[10] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021. 3

[11] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. Deep self-taught learning for weakly supervised object localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1377–1385, 2017. 2

[12] Jeesoo Kim, Junsuk Choe, Sangdoo Yun, and Nojun Kwak. Normalization matters in weakly supervised object localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3427–3436, 2021. 3, 5

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2

[14] 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*, pages 2117–2125, 2017. 2, 3

[15] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 774–782, 2021. 6

[16] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on*

[1] http://host.robots.ox.ac.uk/pascal/VOC/voc2007/

[2] https://pytorch.org/vision/main/generated/torchvision.datasets.VOCDetection.html

[3] http://host.robots.ox.ac.uk/pascal/VOC/voc2007/htmldoc/index.html

[4] https://github.com/petrapostelnicu/ort_cam_wsod

[5] https://www.nwo.nl/en/netherlands-code-conduct-research-integrity

[6] https://github.com/ppengtang/pcl.pytorch/tree/master

pattern recognition (ICPR'06), pages 850–855. IEEE, 2006. 8

[17] 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. 1, 3

[18] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 8

[19] Christopher J Rapson, Boon-Chong Seet, M Asif Naeem, Jeong Eun Lee, Mahmoud Al-Sarayreh, and Reinhard Klette. Reducing the pain: A novel tool for efficient ground-truth labelling in images. In *2018 international conference on image and vision computing New Zealand (IVCNZ)*, pages 1–9. IEEE, 2018. 1

[20] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 8

[21] 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. 1

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2, 3, 5, 8

[23] Christoph Sager, Christian Janiesch, and Patrick Zschech. A survey of image labelling for computer vision applications. *Journal of Business Analytics*, 4(2):91–110, 2021. 1

[24] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3

[25] Feifei Shao, Long Chen, Jian Shao, Wei Ji, Shaoning Xiao, Lu Ye, Yueting Zhuang, and Jun Xiao. Deep learning for weakly-supervised object detection and localization: A survey. *Neurocomputing*, 496:192–207, 2022. 1

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[28] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2843–2851, 2017. 1

[29] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. PCL: Proposal cluster learning for weakly supervised object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 1, 2, 6, 8

[30] 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: 154–171, 2013. 2

[31] Jun Wei, Qin Wang, Zhen Li, Sheng Wang, S Kevin Zhou, and Shuguang Cui. Shallow feature matters for weakly supervised object localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5993–6001, 2021. 1, 3

[32] Xiaopeng Zhang, Jiashi Feng, Hongkai Xiong, and Qi Tian. Zigzag learning for weakly supervised object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4262–4270, 2018. 8

[33] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1325–1334, 2018. 1

[34] Yongqiang Zhang, Yancheng Bai, Mingli Ding, Yongqiang Li, and Bernard Ghanem. W2f: A weakly-supervised to fully-supervised framework for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 928–936, 2018. 2

[35] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014. 2

[36] 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. 1, 2

[37] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 391–405. Springer, 2014. 2

[38] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276, 2023. 1, 2

Object Roughly There: CAM - based Weakly Supervised Object Detection

Reducing the labelling efforts for deep learned object detectors

Supplementary Material

7. Results on 224x224 images

While object detectors benefit from full image sizes, smaller images offer much faster training and testing times, with the disadvantage of losing more fine-grained features. From the experiments in Sec. 4 it can be seen that when creating bounding boxes for large objects, ORT suffers from identifying only the most discriminative features of objects. Under the intuition that smaller image sizes that have less fine-grained features would aid the proposed model in looking at the full extent of larger objects, experiments were performed on the VOC 2007 dataset with all training and testing images resized to 224x224.

Classification: The classification results in Tab. 6 compared to Tab. 1 show that the classification performance of the FPN classifier slightly increases (by approximately 0.4% to 1.2%) when using smaller image sizes, while VGG16 performs sightly worse (by 3.5%). Overall the classifiers still manage to detect the objects in an image, with their Multilabel Average Precision ranging between 80.3% and 87.5%.

Method	Full dataset	Localization			Multi Instance	Multi Class
		small	large	both		
VGG16	80.3	65.8	84.7	82.0	75.9	74.2
FPN P5	86.9	75.3	92.7	89.9	85.9	81.0
FPN P4	87.8	77.4	92.1	89.8	87.3	82.2
FPN P3	86.4	71.9	93.1	89.3	88.3	80.3
FPN P2	87.5	75.3	93.0	90.4	87.0	82.1

Table 6. Classification Multilabel Average Precision on VOC 2007 with images resized to 224x224 of the VGG16 and the FPN based classifier when using the 4 different feature maps. The classifiers used in the proposed models manage to effectively identify the objects present in an image.

Bounding Box detection: When comparing the bounding box detection capabilities of ORT, Tab. 7 and Tab. 2 show that resizing images gives overall better results. Upon further inspection, this is due to the performance increase (by approximately 13.1% to 39.2%) on the large localization data split that the 224x224 images bring. However, there is a significant performance decrease (by approximately 8.8% to 20.7%) in the localization of small objects compared to using full image sizes. It is worth noting that these differences are larger for the FPN-based models than for ORT-VGG16. Additionally, the two stage method has an overall better performance when using the full image size, despite the one stage method having better overall performance when using the resized images.

Method	Full dataset	Localization			Multi Instance	Multi Class
		small	large	both		
ORT-VGG16	11.7	15.8	32.0	26.7	6.2	10.9
+Faster-RCNN	22.3	30.3	51.7	44.0	13.6	20.7
ORT-FPN P5	11.8	0.8	52.9	36.9	5.8	10.2
+Faster-RCNN	17.0	2.4	59.1	41.6	8.5	14.4
ORT-FPN P4	13.6	5.4	54.3	38.8	4.8	11.9
+Faster-RCNN	19.9	11.9	63.2	45.7	11.4	17.6
ORT-FPN P3	14.1	3.1	56.8	39.0	5.1	12.9
+Faster-RCNN	19.2	5.4	65.4	46.2	9.3	16.5
ORT-FPN P2	12.1	4.7	47.1	33.9	3.9	10.5
+Faster-RCNN	19.4	5.6	62.5	44.9	9.3	16.6
Faster-RCNN	70.2	82.8	80.2	82.7	62.4	66.2

Table 7. Object Detection results with mAP@50 on VOC 2007 with images resized to 224x224. From top to bottom: the proposed weakly supervised models using the VGG16 classifier and the FPN classifier with different feature maps, where '+' indicates the two stage method. At the bottom, the SOTA weakly supervised MIL method PCL, where * indicates the result is taken from the original paper and the fully supervised Faster-RCNN. Weakly supervised models are outperformed by the fully supervised detector, struggling most with small objects, multi instance and multi class images. The two stage method boosts the performance of the one stage models.

Figure 6 shows the qualitative detection results on the same images as in Fig. 5, this time resized to 224X224. It can be observed that the resized images aid the models in looking at the full extent of objects when they are large, but it also makes them look at the areas surrounding small objects. Not being able to see fine-grained features, also causes ORT to not be able to separate multiple instances as good as it did in full sized images.

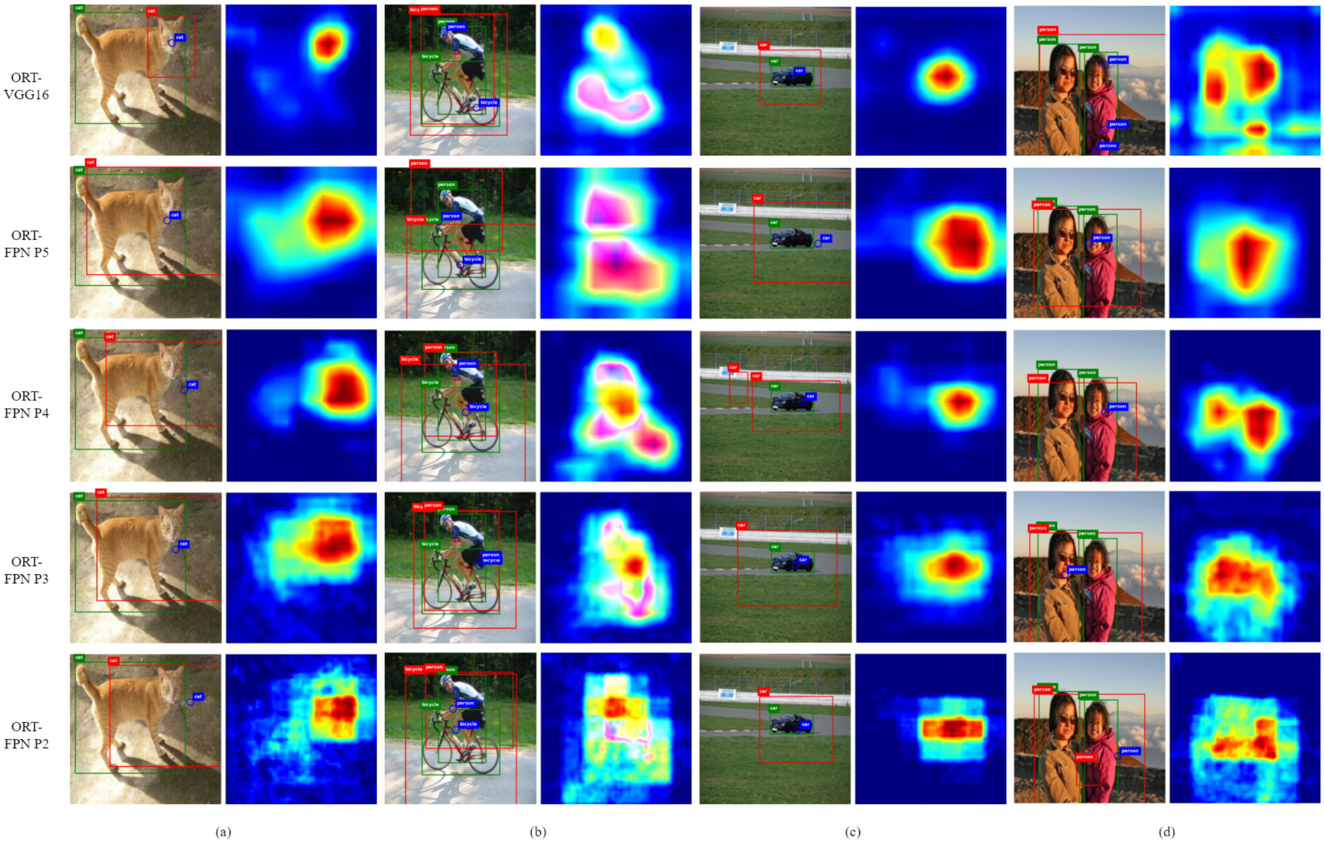

Figure 6. Comparison between the detection and pin pointing performance between the different backbone classifiers used for ORT on a a) large localization image, b) multi instance image, c) small localization image and d) multi class image of the VOC 2007 test set with images resized to 224x224. Each column contains the original image with the ground truth bounding boxes in green, the predicted bounding boxes in red, the predicted pin points in blue and the heatmap generated with GradCAM++. Note that the heatmaps for multi class images are shown by overlaying the heatmaps from the different classes for visualization purposes. The FPN-based models manage to detect more of the objects, compared to the VGG16 that mainly looks at the most discriminative parts. The deeper feature maps in the FPN detect more fine-grained features and have a less uniform aspect. Note that for small objects, the proposed models also detect the surrounding area.

Correct Localization: The CorLoc performance with the resized images can be observed in Tab. 8. Compared to that of the full sized images in Tab. 3, the same trends as with the bounding box detection performance apply, with the exception that the differences are smaller.

Pin Pointing: Looking at the pin pointing performance of ORT with the resized images in Tab. 9 compared to that with the full sized images in Tab. 4, we can see that it performs better when the images are not resized. While its localization capabilities remain approximately the same, when having the full image it performs significantly better (by 6.9% to 15%) on images with multiple instances and multiple classes.

Method	Full dataset	Localization			Multi Instance	Multi Class
		small	large	both		
ORT-VGG16	26.0	27.6	33.4	30.3	17.5	25.3
ORT-FPN P5	28.0	2.4	59.2	42.9	19.0	22.5
ORT-FPN P4	28.3	12.0	51.8	39.4	19.3	23.1
ORT-FPN P3	27.3	6.7	53.0	38.5	18.2	21.6
ORT-FPN P2	23.0	4.3	42.9	32.0	16.5	18.4

Table 8. CorLoc results on the VOC 2007 trainval set with images resized to 224x224 of the proposed one stage method with the VGG16 and the FPN based classifier when using the 4 different feature maps. The proposed models struggle at detecting the full extent of objects. The FPN backbone performs overall better than the VGG16 one.

Method	Full dataset	Localization			Multi Instance	Multi Class
		small	large	both		
ORT-VGG16	68.8	73.0	98.7	95.9	44.5	60.1
+Faster-RCNN	73.2	80.9	91.8	91.3	57.5	67.4
ORT-FPN P5	64.4	65.3	97.8	93.8	44.2	57.1
+Faster-RCNN	68.2	68.0	97.7	94.9	51.2	60.9
ORT-FPN P4	72.9	70.0	99.4	96.3	49.9	66.2
+Faster-RCNN	74.8	78.3	97.9	95.6	56.2	68.8
ORT-FPN P3	72.9	66.4	99.3	96.4	49.3	64.5
+Faster-RCNN	64.5	72.4	94.9	92.5	44.1	58.4
ORT-FPN P2	79.2	72.7	99.4	97.3	55.5	70.1
+Faster-RCNN	68.3	70.9	93.6	90.9	49.3	62.8
Faster-RCNN	95.2	88.7	98.4	98.4	82.9	90.9

Table 9. Pin Pointing results with mAP on VOC 2007 with images resized to 224x224, where a point is considered correct if it falls in the ground-truth bounding box. From top to bottom: the proposed weakly supervised models using the VGG16 classifier and the FPN classifier with different feature maps, where '+' indicates the two stage method. At the bottom, the fully supervised Faster-RCNN. The proposed models can successfully pin point the general location of objects, their performance being close to the fully supervised detector.

8. Qualitative Results

Figure 7, 8, 9 and 10 provide further qualitative comparisons of the different backbone classifier of ORT applied to the VOC 2007 test set. Results show that ORT makes contextual mistakes, meaning that for a predicted class it can detect not only the ground-truth but also surrounding objects that belong to the same context. Furthermore, it has problems with detecting partially occluded objects, as well as objects of the same class that are close together. Overall, FPN backbones show improved granularity in detecting fine features. However, they require specific segmentation threshold tuning to bring out their full potential.

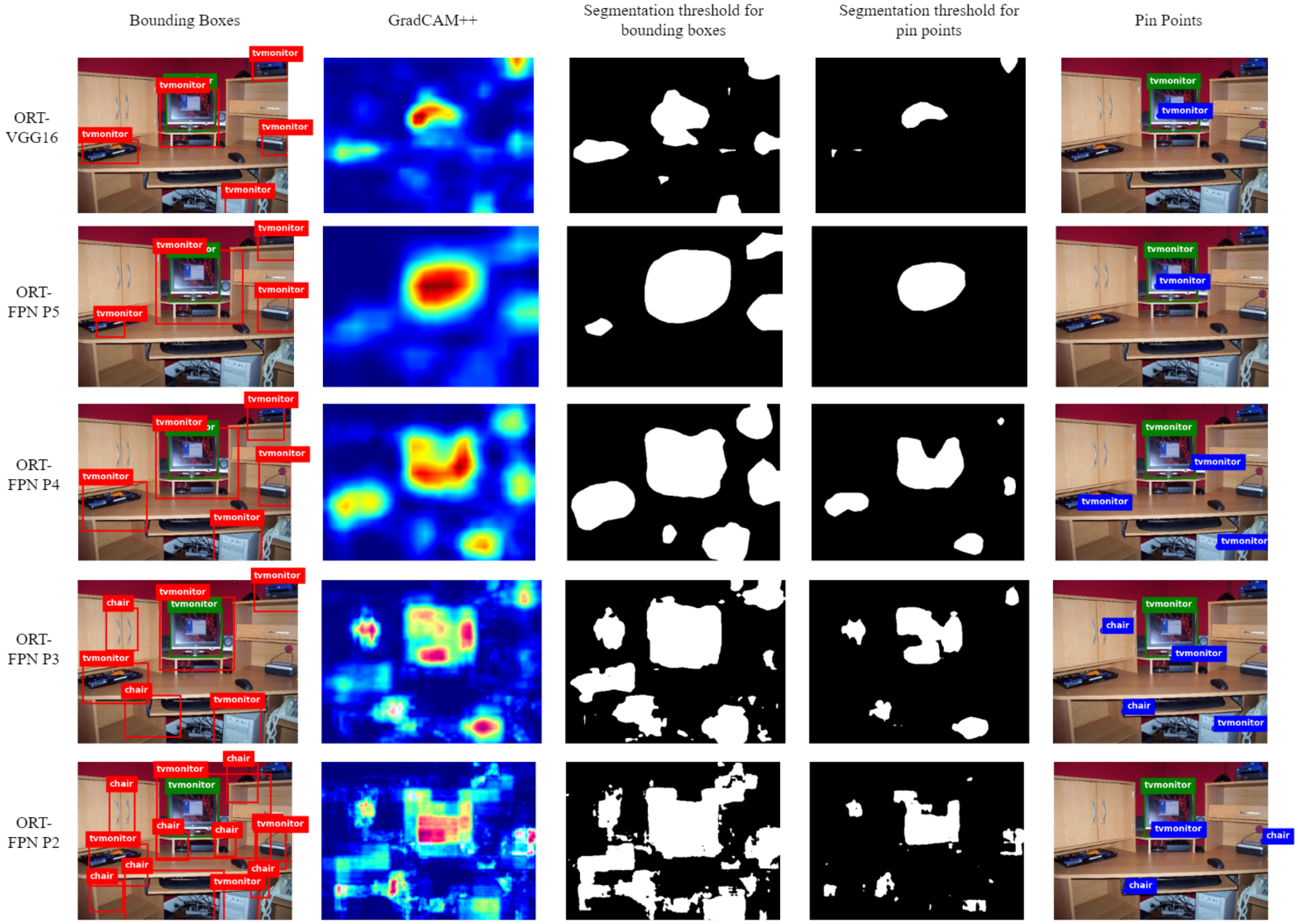

Figure 7. Qualitative results on a localization image of the VOC 2007 test set, with the different backbone classifier architectures. Overall, the ORT makes contextual mistakes, meaning that aside from the object of interest (tv monitor) it detects other objects around that belong to the same context (keyboard, PC).

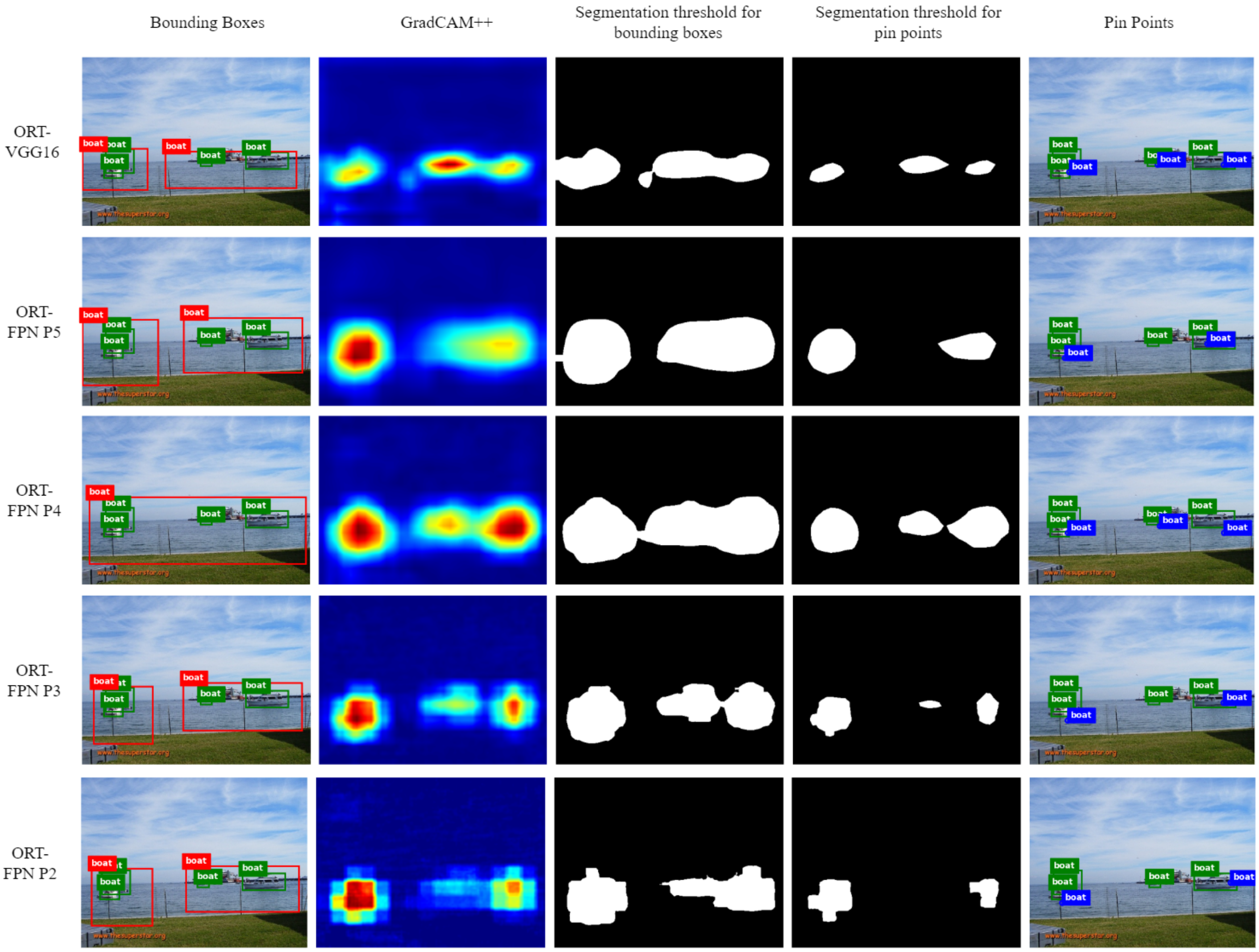

Figure 8. Qualitative results on a multiple instance image of the VOC 2007 test set, with the different backbone classifier architectures. Note that the boat that ORT never detects is partially occluded by another boat. The FPN P3 feature map best localizes the objects, however the thresholds applied would need to be tuned specifically for this layer to leverage its full performance. The other backbone classifier architectures focus less on fine-grained features, leading them to merge together the boats that are near each other.

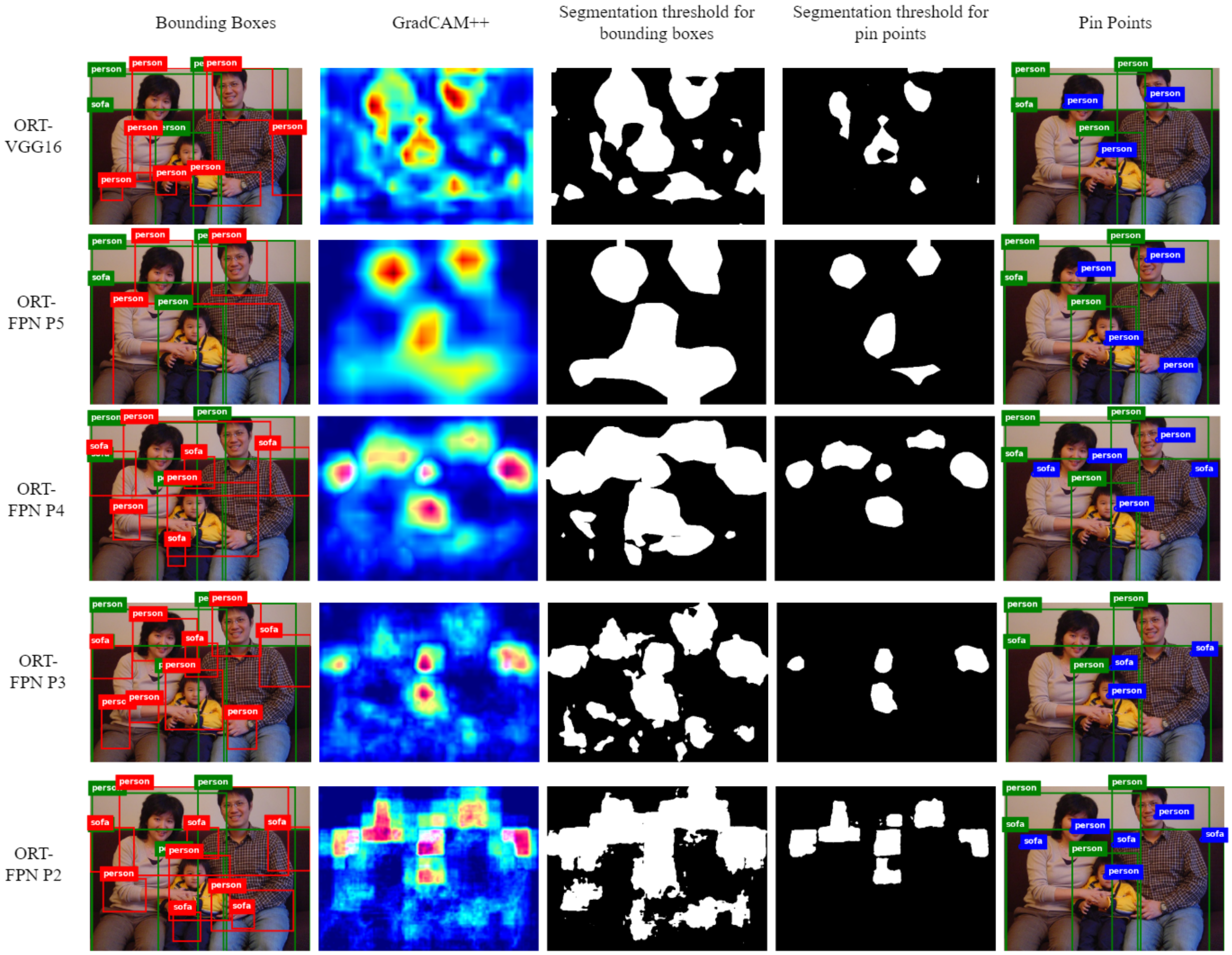

Figure 9. Qualitative results on a multi class multi instance image of the VOC 2007 test set, with the different backbone classifier architectures. While the VGG16 and FPN P5 manage to only detect the people in the image, the subsequent FPN feature maps see more fine grained features, which allows them to detect the sofa. However, the sofa is not detected in the parts where it's occluded by the humans.

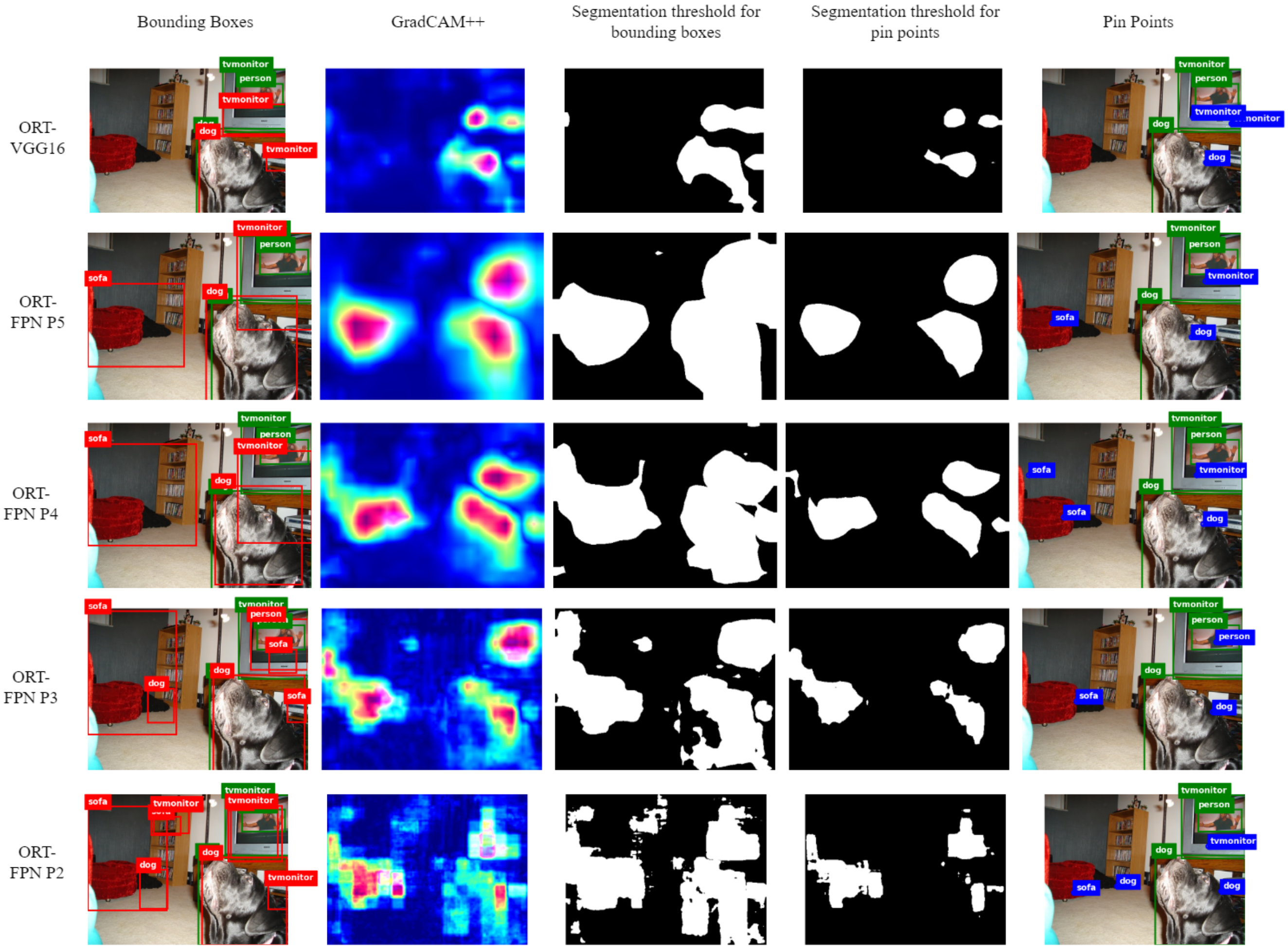

Figure 10. Qualitative results on a multi class image of the VOC 2007 test set, with the different backbone classifier architectures. While the VGG16 only detects the dog and the tv monitor, the FPN classifier also sees the sofa. Note that while the sofa is not marked as ground-truth, this can be a considered a contextual mistake. The deeper FPN feature maps detect the person inside the tv monitor, but don't see the tv monitor surrounding it.