

# Weakly-supervised Localization of Multiple Objects in Images using Cosine Loss

Björn Barz<sup>a</sup> and Joachim Denzler<sup>b</sup>

Computer Vision Group, Friedrich Schiller University Jena, Jena, Germany

**Keywords:** Weakly-supervised Localization, Class Activation Maps, Dense Class Maps, Cosine Loss, Object Detection.

**Abstract:** Can we learn to localize objects in images from just image-level class labels? Previous research has shown that this ability can be added to convolutional neural networks (CNNs) trained for image classification post hoc without additional cost or effort using so-called class activation maps (CAMs). However, while CAMs can localize a particular known class in the image quite accurately, they cannot detect and localize instances of multiple different classes in a single image. This limitation is a consequence of the missing comparability of prediction scores between classes, which results from training with the cross-entropy loss after a softmax activation. We find that CNNs trained with the cosine loss instead of cross-entropy do not exhibit this limitation and propose a variation of CAMs termed Dense Class Maps (DCMs) that fuse predictions for multiple classes into a coarse semantic segmentation of the scene. Even though the network has only been trained for single-label classification at the image level, DCMs allow for detecting the presence of multiple objects in an image and locating them. Our approach outperforms CAMs on the MS COCO object detection dataset by a relative increase of 27% in mean average precision.


## 1 INTRODUCTION


Obtaining annotations for object detection tasks is costly and time-consuming. The largest object detection dataset to date comprises 1.9 million images with 600 object classes (Kuznetsova et al., 2020) and the most popular one merely 120,000 images with 80 classes (Lin et al., 2014). Datasets with image-level class labels, in contrast, are orders of magnitudes larger, containing 9.2 million (Kuznetsova et al., 2020), 18 million (Wu et al., 2019), or even 300 million images with up to 18,000 classes (Sun et al., 2017). This scale could be achieved thanks to semi-automatic acquisition of labeled images through search engines, which is not possible for bounding box annotations. However, what if we could learn object detection models from image-level class labels? This task, where the model prediction is more complex than the supervision signal, is known as *weakly-supervised localization (WSL)*.

Zhou et al. (2016) found that modern convolutional neural network (CNN) classifiers learn this task

implicitly and can be augmented with object localization capabilities post hoc without additional training cost. Modern CNN architectures typically apply global average pooling between the last convolutional and the fully-connected classification layer (He et al., 2016). Zhou et al. remove this pooling operation and apply the weights of the classifier to each cell of the last convolutional feature map individually to obtain a so-called *class activation map (CAM)* for a certain class of interest, e.g., the one predicted by the global classifier. However, CAMs for different classes are not comparable with each other due to different ranges of predicted class scores. This is a consequence of the cross-entropy loss with softmax activation that is typically used for training.

We show that it becomes possible to generate such an activation map for *all* classes that are present in the image at once instead of only for the top-scoring class, when the cross-entropy loss is replaced with the cosine loss during training. This loss function has previously been used successfully for deep learning on small data sets (Barz and Denzler, 2020) and for integrating prior knowledge about the semantic similarity of classes (Barz and Denzler, 2019). The more homogenous classification scores learned by this ob-

<sup>a</sup>  <https://orcid.org/0000-0003-1019-9538>

<sup>b</sup>  <https://orcid.org/0000-0002-3193-3300>

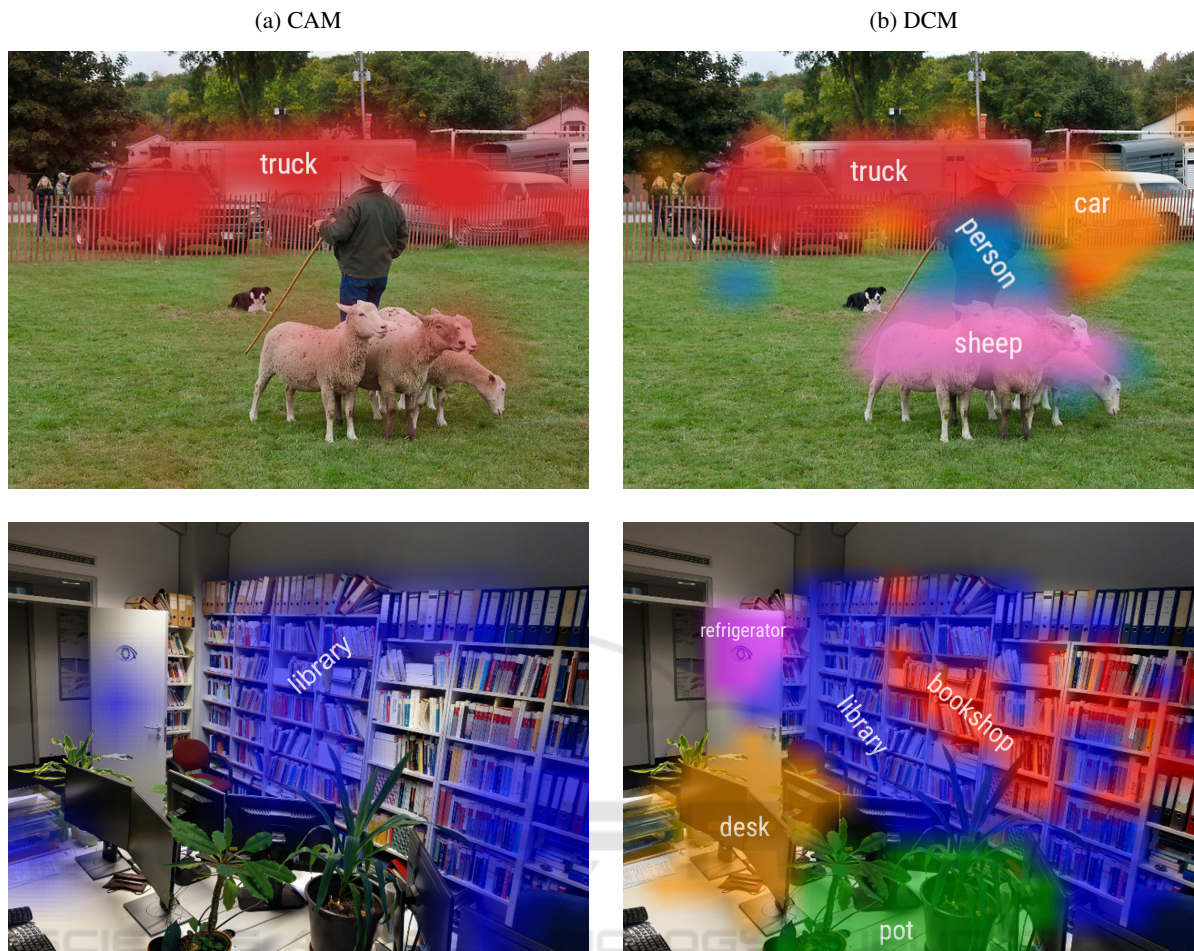


Figure 1: Exemplary comparison of localization results obtained with CAMs (left) and our DCMs (right). The upper example is from the MS COCO dataset (Lin et al., 2014) and both ResNet-50 models were trained on cropped instances from COCO. The lower example shows a photo of our office and uses models trained on ImageNet-1k (Russakovsky et al., 2015).

jective allow us to choose a constant global threshold across all classes for determining whether an object is present at each location in the feature map and what type of object it is. The resulting *dense class map* (DCM) resembles a coarse semantic segmentation. Fig. 1 illustrates this with two examples. Even though the network has only been trained to assign a single class label to an entire image as a whole, it can be modified to locate a variety of objects in a complex scene. We describe our approach in detail in Section 3, after briefly reviewing CAMs Section 2.

In Section 4, we present quantitative and qualitative experimental results on the ILSVRC 2012 Localization Challenge (Russakovsky et al., 2015) and the MS COCO object detection task (Lin et al., 2014). We find that DCMs match the performance of CAMs if the image only contains a single dominant object to be localized, but largely outperform them for detecting multiple objects in a single scene.

Finally, we outline related work in Section 5 and summarize our conclusions in Section 6.

## 2 BACKGROUND: CAMS

Class activation maps (CAMs) have been introduced by Zhou et al. (2016) as a technique for enhancing a CNN pre-trained for image-level classification with object localization capabilities after the training. Since our dense class maps (DCMs) build up on this approach, we briefly review CAMs in the following.

Let  $\psi : \mathbb{R}^{\mathcal{H} \times \mathcal{W} \times 3} \rightarrow \mathbb{R}^{\mathcal{H}' \times \mathcal{W}' \times D}$  denote a feature extractor realized by a CNN up to the last convolutional layer. Given an image with height  $\mathcal{H}$  and width  $\mathcal{W}$ ,  $\psi$  computes a spatial feature map consisting of  $\mathcal{H}' \times \mathcal{W}'$  local feature vectors of dimensionality  $D$ . Due to pooling, the size of this feature map is typically only a fraction of the original image size.

In most modern CNN architectures, e.g., ResNet (He et al., 2016), these local feature cells are averaged into a single feature vector, which is passed through a classification layer with weight matrix  $W \in \mathbb{R}^{C \times D}$  and bias vector  $b \in \mathbb{R}^C$ , where  $C$  denotes the number of classes. The resulting class scores are finally mapped into the space of probability distributions using the softmax activation function:

$$\text{softmax}(y)_i := \frac{\exp(y_i)}{\sum_{j=1}^C \exp(y_j)}. \quad (1)$$

Formally, the predicted probabilities for an input image  $x$  are obtained as:

$$f(x) = \text{softmax} \left( W \cdot \left( \frac{\sum_{i,j} \Psi(x)_{i,j}}{\mathcal{H}' \cdot \mathcal{W}'} \right) + b \right). \quad (2)$$

The key insight of Zhou et al. (2016) is that, due to linearity, the classifier can also be applied to each local feature cell before pooling without changing the result:

$$f(x) = \text{softmax} \left( \frac{\sum_{i,j} W \cdot \Psi(x)_{i,j} + b}{\mathcal{H}' \cdot \mathcal{W}'} \right). \quad (3)$$

The predicted global logits are hence the average of region-wise class scores. These local scores for any  $c \in \{1, \dots, C\}$  are the class activation map of class  $c$ :

$$\text{CAM}(x)_{i,j,c} = W_c \cdot \Psi(x)_{i,j} + b_c. \quad (4)$$

Even though the classifier has only been trained on image-wise labels, Zhou et al. (2016) found that the CAM of the top-scoring class can effectively localize instances of that class. To this end, they threshold the CAM by 20% of its maximum and draw a bounding box around the largest connected component.

## 3 METHOD

### 3.1 Cosine Loss

The relative thresholding approach of CAMs hints at one of their main issues: Before the softmax activation, the ranges of the CAMs for different classes are usually not comparable. This complicates setting a single threshold for deciding whether a certain object is present at a given location in the image or not (see Fig. 2a). If the softmax activation would be applied—which CAMs do not—distinguishing between objects and noisy background patches became an issue, since these could reach similarly high activation values due to the softmax operation (see Figs. 2c and 2e).

These problems do not exist when using the cosine loss (Barz and Denzler, 2020) for training, which

enables us to use a single absolute threshold for the simultaneous detection of multiple classes instead of a single one. Instead of the softmax function, the cosine loss applies  $L^2$  normalization to the feature representation  $\hat{x} \in \mathbb{R}^{D'}$  of the input computed by the network and maximizes its cosine similarity to the embedding  $\varphi(c)$  of the ground-truth class  $c \in \{1, \dots, C\}$ :

$$\mathcal{L}_{\text{cos}}(\hat{x}, c) := 1 - \frac{\langle \hat{x}, \varphi(c) \rangle}{\|\hat{x}\| \cdot \|\varphi(c)\|}. \quad (5)$$

The class embeddings  $\varphi(c)$  can be derived from prior semantic knowledge such as class taxonomies (Barz and Denzler, 2019), from world knowledge encoded in large text corpora (Frome et al., 2013), or simply be one-hot encodings. In the latter case, the cosine loss maximizes the  $c$ -th entry of the prediction vector after  $L^2$  normalization. Compared to cross-entropy with softmax, it enforces this channel less strictly to become 1.

It can be seen in Fig. 2d that foreground and background are much better separated in the histogram of maximum class scores after applying the activation. The  $L^2$  normalization also accounts for making the responses at different locations comparable by discarding the magnitude of the difference between predicted features and class embeddings and focusing on their angle instead.

### 3.2 Dense Class Maps

Leveraging these advantages of the cosine loss, we build upon the idea of CAMs and obtain a dense map of class embeddings DCE:  $\mathbb{R}^{\mathcal{H} \times \mathcal{W} \times 3} \rightarrow \mathbb{R}^{\mathcal{H}' \times \mathcal{W}' \times D'}$  for an image  $x \in \mathbb{R}^{\mathcal{H} \times \mathcal{W} \times 3}$  by removing the global average pooling layer and converting the embedding layer with weights  $W \in \mathbb{R}^{D' \times D}$  and biases  $b \in \mathbb{R}^{D'}$  into a  $1 \times 1$  convolution. The entire procedure is depicted schematically in Fig. 3.

In contrast to CAMs, which do not apply the softmax activation on the local predictions, we apply the  $L^2$  normalization at each cell of the resulting tensor:

$$\text{DCE}(x)_{i,j} = \frac{W \cdot \Psi(x)_{i,j} + b}{\|W \cdot \Psi(x)_{i,j} + b\|_2}. \quad (6)$$

Averaging over all locations of the resulting dense embedding maps is, hence, not equivalent anymore to the output of the original network.

We can then assign a label  $\text{class}(x, i, j) \in \{1, \dots, C\}$  to each local cell by finding the class embedding that is most similar to its local feature vector:

$$\text{sim}(x, i, j, c) = \langle \text{DCE}(x)_{i,j}, \varphi(c) \rangle, \quad (7a)$$

$$\text{class}(x, i, j) = \underset{c=1, \dots, C}{\text{argmax}} \text{sim}(x, i, j, c). \quad (7b)$$

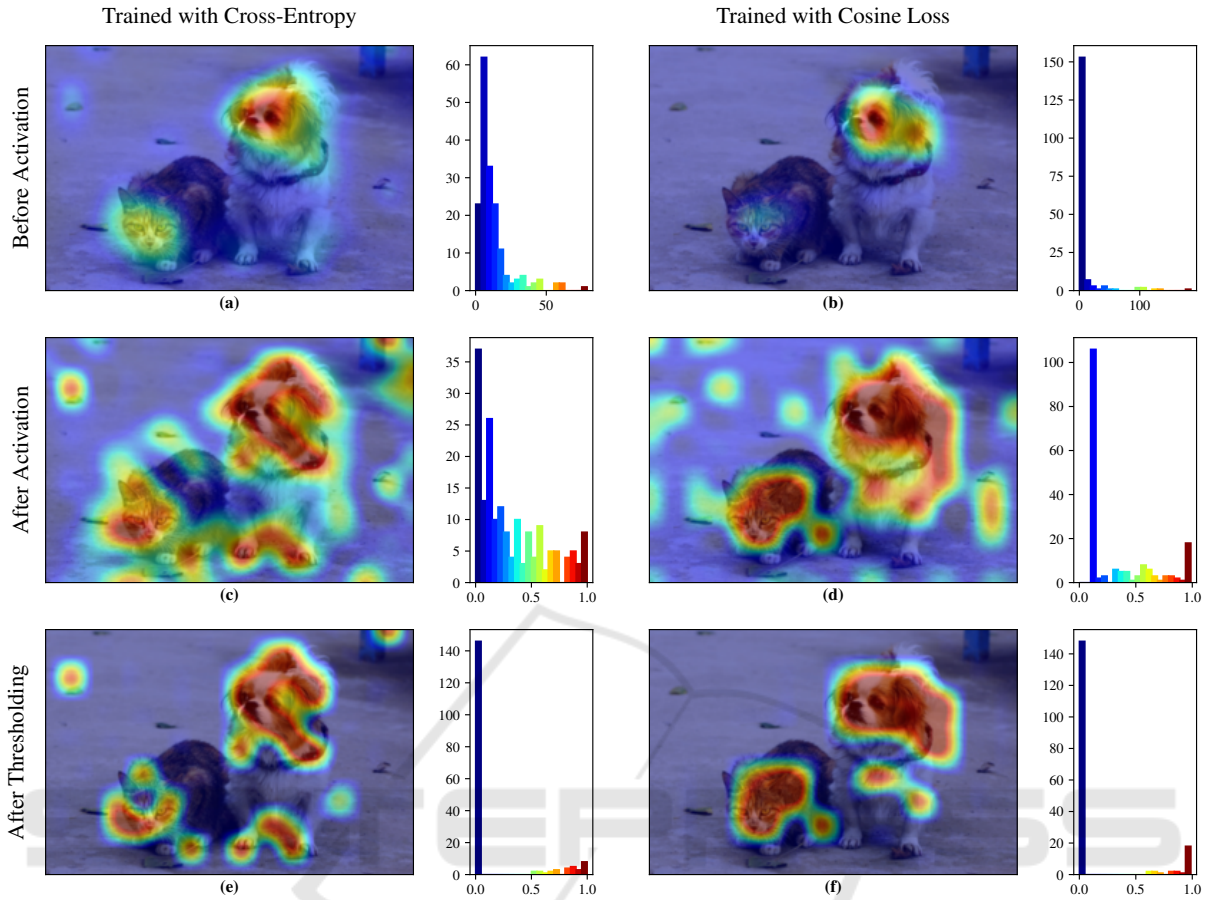


Figure 2: Heatmaps and histograms of the maximum value over all classes at each cell of a DCM, before and after the activation. The activation function is softmax for the cross-entropy loss and  $L^2$  normalization for the cosine loss. The hyper-parameters for thresholding are  $\vartheta_{\text{cls}} = 0.8$ ,  $\vartheta_{\text{sim}} = 0.5$ .

However, many of these cells will contain background, such that assigning an object class to them is not reasonable and the results will be noisy. Thus, we first determine a set  $\mathcal{C}(x) \subseteq \{1, \dots, C\}$  of classes that are present in the image by assigning a score to each class and selecting those classes whose score is greater than a certain threshold  $\vartheta_{\text{cls}} \in [0, 1]$ , i.e.,  $\mathcal{C}(x) = \{c \in \{1, \dots, C\} \mid \text{score}(x, c) > \vartheta_{\text{cls}}\}$ . The score for a given class is defined as the maximum cosine similarity to its class embedding over all locations to which this class has been assigned:

$$\text{score}(x, c) = \max_{i, j} \mathbb{1}_{\{\text{class}(x, i, j) = c\}} \cdot \text{sim}(x, i, j, c), \quad (8)$$

where  $\mathbb{1}_{\{\cdot\}}$  is the indicator function, being one if the argument evaluates to true and zero otherwise. This scoring procedure is different from simply selecting the globally top-scoring classes, since two related classes could obtain high scores at identical positions. With our approach, only the highest-scoring of such overlapping classes will be selected.

For finding the locations where the selected classes are present, we threshold their cosine similarity with a second threshold  $\vartheta_{\text{sim}} \leq \vartheta_{\text{cls}}$  to obtain a *dense class map*  $\text{DCM}(x, i, j) \in \{0, 1, \dots, C\}$ , where 0 denotes the background class:

$$\text{DCM}(x, i, j) = \begin{cases} \text{class}(x, i, j) & \text{if } \text{class}(x, i, j) \in \mathcal{C} \text{ and} \\ & \text{sim}(x, i, j, \text{class}(x, i, j)) > \vartheta_{\text{sim}}, \\ 0 & \text{else.} \end{cases} \quad (9)$$

An example of the similarity scores corresponding to the resulting class maps after this two-stage thresholding procedure—first across classes, then across locations—is given in Fig. 2f. The actual output that is relevant for practical use, however, is the hard assignment of locations to classes given by the DCM. This can be visualized by color-coding classes as done in Fig. 1.

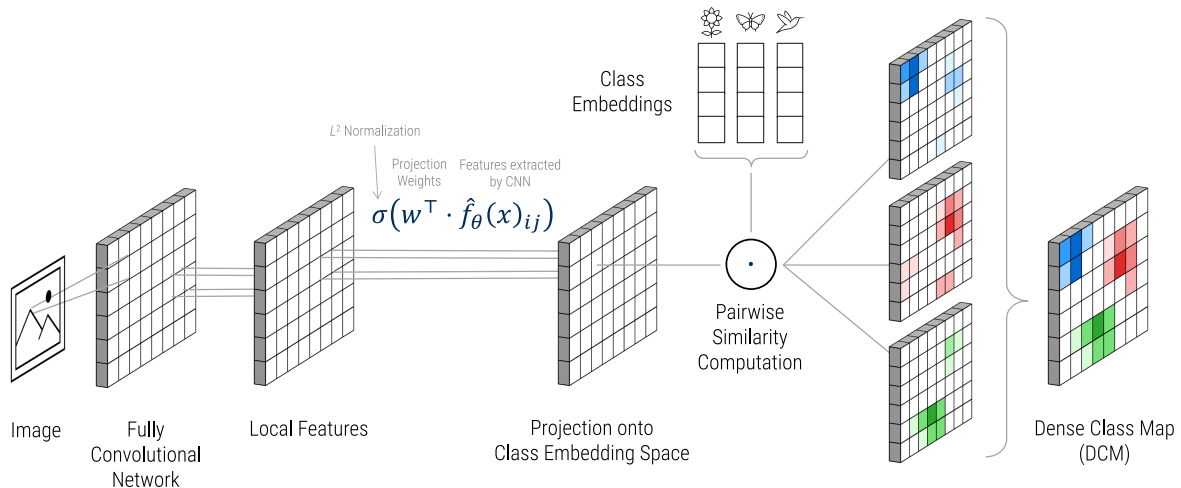


Figure 3: Schematic illustration of the pipeline for computing a dense class map.

### 3.3 Implementation Details

While training images annotated with a single class usually show a close-up of the object of interest, we are interested in analyzing more complex scenes where the objects are smaller. Thus, we use a higher input image resolution for generating DCMs than the resolution typically used for training the network, which usually is  $224 \times 224$  for a ResNet trained on ImageNet (He et al., 2016). For dense classification, we resize the input images so that their larger side is 640 pixels wide.

It is worth noting that the resolution of the feature map obtained from the last convolutional layer, and hence the resolution of the DCM as well, is rather coarse. In the case of the ResNet-50 architecture, the dimensions of the DCM are  $\frac{1}{32}$  of the original image dimensions. For visualization purposes, we upsample the DCM and all heatmaps to the size of the image using bicubic interpolation, which results in a rather blurry semantic segmentation.

For generating bounding boxes instead of segmentations, we generate one box for each class in  $\mathcal{C}$  following the approach of Zhou et al. (2016): The similarity map  $\text{sim}(x, \cdot, c)$  for class  $c \in \mathcal{C}$  is thresholded with  $\mathfrak{d}_{\text{bbox}} \cdot \max_{i,j} \text{sim}(x, i, j, c)$ , where  $\mathfrak{d}_{\text{bbox}} \in [0, 1]$ , and a box is drawn around the largest connected component.

## 4 EXPERIMENTS

We evaluate our DCM approach in comparison to CAMs in two settings: First, we conduct WSL on the ImageNet-1k dataset (Russakovsky et al., 2015),

which mainly comprises images showing a single dominant object and provides a single class label and corresponding bounding box per image. This experiment serves as a verification that DCMs do not perform worse than CAMs in this restricted setting, in which CAMs typically operate. Second, we use the MS COCO dataset (Lin et al., 2014), an established benchmark for object detection, for evaluating both approaches in a more realistic setting where multiple different classes are present in a single image.

### 4.1 Semantic Information

As mentioned in Section 3.1, using the cosine loss for training allows for straightforward integration of prior knowledge in the form of semantic class embeddings. In addition to one-hot encodings, we evaluate DCMs on networks trained with the hierarchy-based semantic embeddings proposed by Barz and Denzler (2019). These embeddings are constructed so that their pairwise cosine similarity equals a semantic similarity measure derived from a class taxonomy.

To obtain such a taxonomy covering the 80 classes of MS COCO, we map them manually to matching synsets from the WordNet ontology (Fellbaum, 1998). However, the WordNet graph is not a tree, because some concepts have multiple parent nodes. Thus, we prune the subgraph of WordNet in question to a tree using the approach of Redmon and Farhadi (2017): We start with a tree consisting of the paths from the root to all classes from MS COCO which have only one such root path. Then, the remaining classes are added successively by choosing that path among their several root paths that results in the least number of nodes added to the existing tree.

## 4.2 Training Details

For all our experiments, we train a ResNet-50 (He et al., 2016) using stochastic gradient descent with a cyclic learning rate schedule with warm restarts (Loshchilov and Hutter, 2017). The base cycle length is 12 epochs and is doubled after each cycle. We use five cycles, amounting to a total of 372 training epochs. The base learning rate is 0.1.

For training on ImageNet-1k, the model is initialized with random weights. These pre-trained weights are used as initialization for training on MS COCO.

Since our training objective is single-label classification but images in MS COCO typically contain multiple objects, we extract crops of each individual object using the provided bounding box annotations, ignoring small objects whose bounding box is smaller than 32 pixels in both dimensions. As a result, we obtain 684,070 training images of object instances.

To present the CNN with objects of various scales as well as truncated objects during training, we resize the training images by choosing the target size of their smaller side from [256, 480] at random and extract a square crop of size  $224 \times 224$ . We furthermore use random horizontal flipping and random erasing (Zhong et al., 2017) as data augmentation.

## 4.3 WSL on ImageNet-1k

As a sanity check, we first evaluate the localization performance of DCMs on the easier task of locating a single predominant object in an image using the ImageNet-1k dataset. Following the evaluation protocol defined for the ImageNet Large Scale Recognition Challenge (ILSVRC) 2012 (Russakovsky et al., 2015), we assess performance in terms of the average top-5 localization error. For a single image, the error is 0 if at least one of the top-5 predicted bounding boxes belongs to the object class assigned to the image and has at least 50% overlap with the ground-truth bounding box of that object. Otherwise, the error is 1.

We obtain the same top-5 localization error of 48% with both CAMs applied to a CNN trained with cross-entropy and with DCMs applied to a CNN trained with the cosine loss and one-hot encodings. For CAMs, this performance was obtained with  $\vartheta_{\text{bbox}} = 0.2$ , while we used  $\vartheta_{\text{bbox}} = 0.06$  for DCMs. Zhou et al. (2016) used different network architectures for their CAM experiments, but the performance reported by them is similar.

Thus, both approaches perform equally well on the object localization task of ILSVRC 2012. This was expected, since ILSVRC is easy in this regard. Most images contain only a single object, which often

fills a large part of the image. Detecting multiple objects from different classes and of smaller size is hence not necessary in this scenario.

## 4.4 WSL on MS COCO

### 4.4.1 Evaluation Metric

The MS COCO dataset poses a much more difficult challenge for WSL by presenting a real detection task. Performance is hence typically evaluated in terms of mean average precision (mAP). COCO uses an average over 20 mAP values with different intersection-over-union (IoU) thresholds varying from 50% to 95%. In this work, however, we are not dealing with a fully supervised scenario and the WSL methods never see ideal bounding boxes during training. Therefore, the predicted bounding box can in many cases be much smaller than the ground-truth bounding box if only a single characteristic part of the object is used for the classification decision (e.g., only the head of a dog instead of the entire body). On the other hand, it can also be larger if many objects of the same class stand close together.

These predictions can, however, still be helpful for determining which objects are located where in the image, even if the localization is not highly accurate. Therefore, we report mAP with a more relaxed IoU threshold of 25%. In addition, we also compute mAP with an IoU threshold of 0%, which does not require any overlap with the ground-truth bounding box at all and hence evaluates multi-label classification performance, where we are only interested in which objects are present in the image, but not where they are.

### 4.4.2 Bounding Box Hyper-parameters

Since average precision summarizes the performance of a detector over all possible detection thresholds, we do not need to fix the class score threshold  $\vartheta_{\text{cls}}$  for this experiment. The bounding box generation threshold, on the other hand, is tuned on the uncropped training set individually for each method and then applied on the test set for the final performance evaluation. This results in  $\vartheta_{\text{bbox}} = 0.25$  for CAMs,  $\vartheta_{\text{bbox}} = 0.3$  for DCMs with one-hot encodings, and  $\vartheta_{\text{bbox}} = 0.75$  for DCMs with semantic class embeddings. The comparatively high threshold in the latter case has an intuitive explanation, since different objects are considered more similar to each other on average with semantic embeddings than with one-hot encodings, requiring a higher threshold to prevent over-detection.

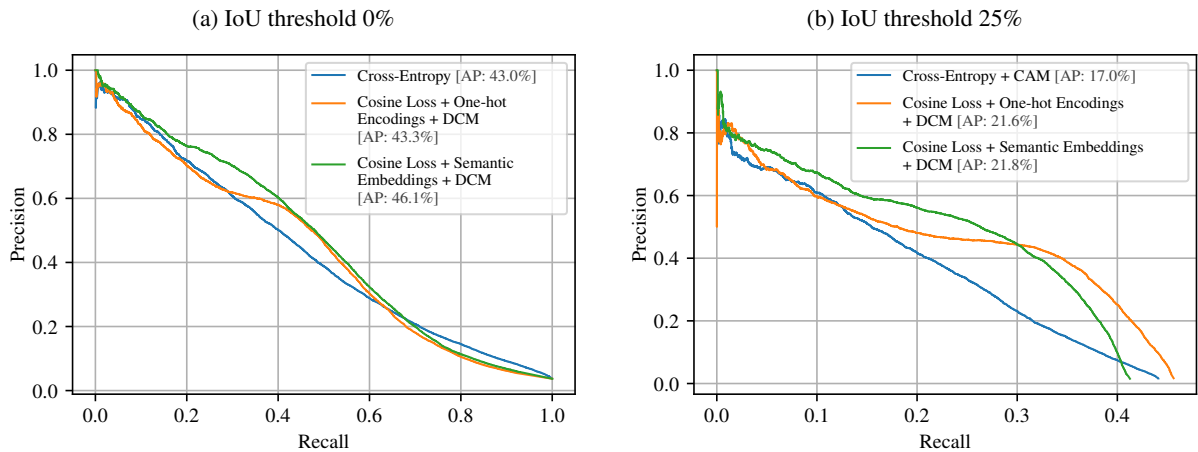


Figure 4: Precision-recall curves for CAMs and DCMs on MS COCO with IoU thresholds of 0% and 25%.

#### 4.4.3 Results

Precision-recall curves and the mAP obtained on the test set are presented in Fig. 4. First, it can be seen that even for the task of multi-label classification without localization (Fig. 4a), the cosine loss combined with DCMs improves mAP slightly compared to networks trained with cross-entropy (43.3% vs. 43.0%). Apparently, suppressing non-maximum class predictions at identical locations—as done by DCMs—helps avoiding false positive predictions. Integrating prior semantic knowledge in the form of hierarchy-based class embeddings improves mAP much further by three percent points.

The differences are more pronounced, though, in an actual object detection setting, as can be seen in Fig. 4b. Using the cosine loss and DCMs allows for maintaining a much higher precision when lowering the detection threshold: For a recall level of 30%, the precision of DCMs with semantic embeddings or one-hot encodings is still 44%, while CAMs applied to a CNN trained with cross-entropy only obtain 23% precision at this level of recall.

As before, semantic embeddings surpass one-hot encodings. They do not obtain the same maximum recall, but provide better precision, i.e., less false positives, for most recall levels. But even without semantic embeddings, DCMs outperform CAMs by a relative increase of 27% mAP.

#### 4.5 Qualitative Examples

Two qualitative examples comparing CAMs and DCMs on ImageNet-1k and MS COCO are shown in Fig. 1. Further examples from MS COCO including bounding boxes are presented in Fig. 5.

For CAMs, we consider all classes with a predicted probability of at least 15% to be present in

the image, since second-best predictions often have low scores due to the softmax operation. Regarding DCMs, we want to select only those classes whose embeddings are highly similar to at least one local feature and hence use the threshold  $\vartheta_{\text{cls}} = 0.99$ . To obtain the coarse semantic segmentations, we set the threshold applied to the similarity maps of selected classes equal to the one used for bounding box generation, i.e.,  $\vartheta_{\text{sim}} = \vartheta_{\text{bbox}}$ , using the bounding box thresholds given above.

The first example in Fig. 5 shows an image comprising objects from two different classes. CAMs are only able to detect one of these correctly due to the strong decision enforced by the softmax activation. The use of the cosine loss and DCMs, on the other hand, allows us to detect both objects in the image using a global threshold across all classes based on the cosine similarity between the predicted features and the class embeddings.

Additionally, integrating prior knowledge about the similarity between classes and hence not forcing the network to consider cars and trucks as two completely different things allows the DCM to also provide the correct prediction “car” along with “truck”, even though the latter has a slightly higher score. Thanks to semantic embeddings, both classes can have high scores simultaneously, since they are semantically similar. Due to the reduced competition between them, not only “truck” exceeds the threshold  $\vartheta_{\text{cls}}$ , but “car” can do so too.

The second example shows a more complex scene, where DCMs, especially with semantic embeddings, are able to detect substantially more objects than CAMs. They do not only detect the train and the bench but also the bicycle, the table, and the person.

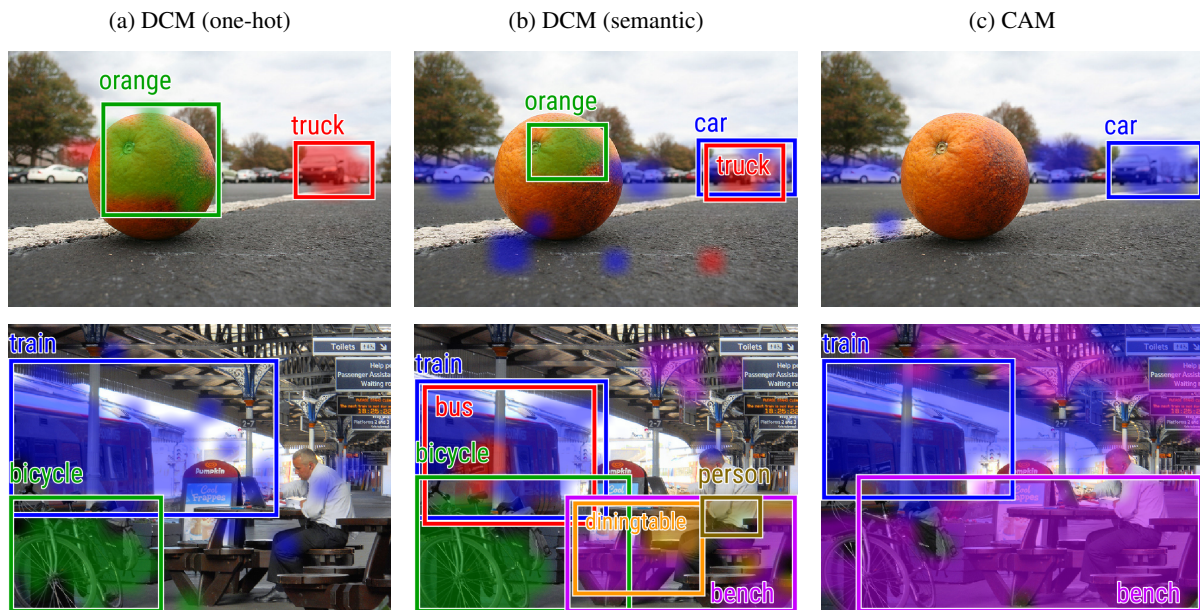


Figure 5: Qualitative examples from MS COCO with bounding boxes.

## 5 RELATED WORK

### 5.1 Class Activation Maps

The applicability of CAMs is restricted by design to CNN architectures with global average pooling followed by a single classification layer. Grad-CAM (Selvaraju et al., 2017) is a generalization of CAMs, which determines weights for the different channels of the convolutional feature map based on the gradient of the network output with respect to the activations in each channel. This allows for obtaining activation maps for arbitrary CNN architectures as well as networks trained for a task different from classification, such as image captioning or question answering.

Gradient-based methods exhibit some drawbacks, though, such as being strongly influenced by the input image and less sensitive to the actual model (Adebayo et al., 2018). Desai and Ramaswamy (2020) avoid these issues by proposing a gradient-free variant of CAMs based on an ablation procedure. This so-called Ablation-CAM determines the weights for the feature channels based on the change of the model output if this channel would be set to zero.

These modifications of the original CAM technique make it more widely applicable to different types of models but do not entail significant advantages over the localization performance of CAMs for ResNet-like classifiers. Most notably, they do not address the limitation of not being able to detect more than one class per image, which we focus on.

### 5.2 Weakly-supervised Localization

Other common issues of CAMs are under- and over-detection: For some classes, CAMs focus only on the most salient parts of the object and ignore the rest, e.g., the heads of persons instead of their body, resulting in too small bounding boxes. Concerning certain other classes such as trains, for example, the classifier often considers context features like rails to be indicative of the object, such that they are mistakenly included in the bounding box.

The latter problem is tackled by *soft proposal networks (SPNs)* (Zhu et al., 2017), which mask the computed feature map with soft objectness scores that are obtained by a random walk on a fully-connected graph over all cells of the feature map. The edge weights of the graph depend on feature differences and spatial distance, such that high objectness scores are assigned to cells that are highly different from their neighbors. These masks are intended to cast a spotlight onto the actual object, reducing the influence of the context. Since the object proposals depend directly on the feature map itself, they are trained jointly with the network.

To mitigate the problem of under-detection, Durand et al. (2017) learn and average multiple CAMs per class. While they intend these maps to highlight different object parts such as heads, legs etc., this property is not explicitly enforced. Zhang et al. (2018) employ a two-branch classifier and use the CAM obtained from the first classifier to erase the



respective object parts from the feature map before passing it through the second classifier, so that this one is forced to focus on different characteristic features of the object. At inference time, the two CAMs are aggregated by taking the element-wise maximum.

As opposed to these approaches, which modify the network architecture with the goal of improving the localization accuracy of CAMs, we do not explicitly aim for improving the state of the art in WSL of single objects. In fact, a recent study found that no WSL method proposed after CAMs actually outperform them in a realistic setting for segmenting images into foreground and background (Choe et al., 2020).

In our work, we extend the WSL approach inspired by CAMs for detecting *multiple* classes at once without needing to tailor the network architecture or learning process to this task. Instead, we find that the simple change of the loss function to the cosine loss for classifier pre-training facilitates generating a joint activation map for all classes present in the image without additional cost.

### 5.3 Cosine Loss

The cosine loss mainly enjoys popularity in the area of multi-modal representation learning and cross-modal retrieval (Sudholt and Fink, 2017; Salvador et al., 2017), where it is used for maximizing the similarity between embeddings of two related samples from different modalities (e.g., images and captions). Similar to aligning representations for different modalities, it has also proven useful for matching learned image representations with semantic class embeddings for semantic image retrieval (Barz and Denzler, 2019). Qin et al. (2008) furthermore used the cosine loss for a list-wise learning to rank approach, where a vector of predicted ranking scores is compared to a vector of ground-truth scores using the cosine similarity.

Barz and Denzler (2020) recently proposed to employ the cosine loss for classification either as a replacement for or in combination with the cross-entropy loss. They found that the  $L^2$  normalization applied by the cosine loss acts as a useful regularizer that improves the classification performance when few training data is available. We follow their work in the sense that we also train CNN classifiers using the cosine loss, but we study the properties of the learned representations from a different perspective, i.e., their advantages for weakly-supervised localization.

## 6 CONCLUSIONS

We proposed an extension of the popular class activation maps (CAMs) for weakly-supervised localization of *multiple* classes in images. The basis of our approach is the use of the cosine loss instead of cross-entropy with softmax for training the classifier on images of individual objects. We found the similarities between local feature cells and class embeddings to be better comparable across different classes than class prediction scores generated by softmax networks, which allows for detecting the presence of multiple classes using a fixed global threshold.

Experiments on the MS COCO dataset showed that our dense class maps (DCMs) improve the object detection performance compared to CAMs by a relative amount of 27% mAP with one-hot encodings and by 28% with semantic class embeddings. At a recall level of 30%, DCMs provide almost twice the precision of CAMs. With semantic embeddings, DCMs do not achieve the same maximum recall as with one-hot encodings, but maintain higher precision.

Even in the easier scenario of multi-label classification without localization, DCMs provide a 0.7% better accuracy. Adding prior knowledge in the form of semantic class embeddings improves the accuracy much further by another 6%.

So far, we relied on the property that images in the dataset used for classification pre-training display a single dominant object. Future work might explore approaches for extending the cosine loss to multi-label classification, so that the pre-training with image-level class labels can be conducted on more generic and complex images.

## REFERENCES

- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018). Sanity checks for saliency maps. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31. Curran Associates, Inc.
- Barz, B. and Denzler, J. (2019). Hierarchy-based image embeddings for semantic image retrieval. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 638–647.
- Barz, B. and Denzler, J. (2020). Deep learning on small datasets without pre-training using cosine loss. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1360–1369.
- Choe, J., Oh, S. J., Lee, S., Chun, S., Akata, Z., and Shim, H. (2020). Evaluating weakly supervised object localization methods right. In *IEEE Conference on Com-*

- puter Vision and Pattern Recognition (CVPR), pages 3130–3139.
- Desai, S. and Ramaswamy, H. G. (2020). Ablation-CAM: Visual explanations for deep convolutional network via gradient-free localization. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 972–980.
- Durand, T., Mordan, T., Thome, N., and Cord, M. (2017). WILDCAT: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5957–5966.
- Fellbaum, C. (1998). *WordNet*. Wiley Online Library.
- Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. (2013). DeViSE: A deep visual-semantic embedding model. In *International Conference on Neural Information Processing Systems (NIPS)*, NIPS’13, pages 2121–2129, USA. Curran Associates Inc.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., Kolesnikov, A., Duerig, T., and Ferrari, V. (2020). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision (IJCV)*.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *European Conference on Computer Vision (ECCV)*, pages 740–755, Cham. Springer International Publishing.
- Loshchilov, I. and Hutter, F. (2017). SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*.
- Qin, T., Zhang, X.-D., Tsai, M.-F., Wang, D.-S., Liu, T.-Y., and Li, H. (2008). Query-level loss functions for information retrieval. *Information Processing & Management*, 44(2):838–855.
- Redmon, J. and Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.
- Salvador, A., Hynes, N., Aytar, Y., Marin, J., Ofli, F., Weber, I., and Torralba, A. (2017). Learning cross-modal embeddings for cooking recipes and food images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3068–3076. IEEE.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 618–626.
- Sudholt, S. and Fink, G. A. (2017). Evaluating word string embeddings and loss functions for CNN-based word spotting. In *International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 493–498. IEEE.
- Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *IEEE International Conference on Computer Vision (ICCV)*.
- Wu, B., Chen, W., Fan, Y., Zhang, Y., Hou, J., Liu, J., and Zhang, T. (2019). Tencent ML-images: A large-scale multi-label image database for visual representation learning. *IEEE Access*, 7:172683–172693.
- Zhang, X., Wei, Y., Feng, J., Yang, Y., and Huang, T. S. (2018). Adversarial complementary learning for weakly supervised object localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1325–1334.
- Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. (2017). Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*.
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929.
- Zhu, Y., Zhou, Y., Ye, Q., Qiu, Q., and Jiao, J. (2017). Soft proposal networks for weakly supervised object localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1859–1868.