This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

# Weakly Supervised Deep Detection Networks

Hakan Bilen
University of Oxford
hbilen@robots.ox.ac.uk

Andrea Vedaldi
University of Oxford
vedaldi@robots.ox.ac.uk

## Abstract

*Weakly supervised learning of object detection is an important problem in image understanding that still does not have a satisfactory solution. In this paper, we address this problem by exploiting the power of deep convolutional neural networks pre-trained on large-scale image-level classification tasks. We propose a weakly supervised deep detection architecture that modifies one such network to operate at the level of image regions, performing simultaneously region selection and classification. Trained as an image classifier, the architecture implicitly learns object detectors that are better than alternative weakly supervised detection systems on the PASCAL VOC data. The model, which is a simple and elegant end-to-end architecture, outperforms standard data augmentation and fine-tuning techniques for the task of image-level classification as well.*

## 1. Introduction

In recent years, Convolutional Neural Networks (CNN) [21] have emerged as the new state-of-the-art learning framework for image recognition. Key to their success is the ability to learn from large quantities of labelled data the complex appearance of real-world objects. One of the most striking aspects of CNNs is their ability to learn generic visual features that are generalise to many tasks. In particular, CNNs pre-trained on datasets such as ImageNet ILSVRC have been shown to obtain excellent results in recognition in other domains [8], in object detection [12], in semantic segmentation [13], in human pose estimation [33], and in many other tasks.

In this paper we look at how the power of CNNs can be leveraged in *weakly supervised detection* (WSD), which is the problem of learning object detectors using only image-level labels. The ability of learning from weak annotations is very important for two reasons: first, image understanding aims at learning an growing body of complex visual concepts (*e.g.* hundred thousands object categories in ImageNet). Second, CNN training is data-hungry. Therefore, being able to learn complex concepts using only light supervision can reduce significantly the cost of data annotation in tasks such as image segmentation, image captioning, or object detection.

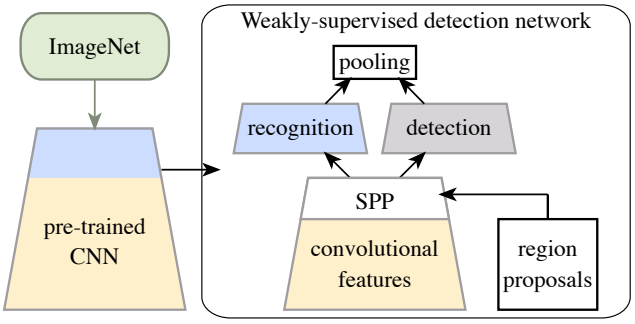


Figure 1. **Weakly Supervised Deep Detection Network.** Our method starts from a CNN pre-trained for image classification on a large dataset, e.g. ImageNet. It then modifies to reason efficiently about regions, branching off a recognition and a detection data streams. The resulting architecture can be fine-tuned on a target dataset to achieve state-of-the-art weakly supervised object detection using only image-level annotations.

We are motivated in our research by the hypothesis that, since pre-trained CNNs generalise so well to a large number of tasks, they should contain meaningful representations of the data. For example, there exists evidence that CNNs trained for image classification learn proxies to objects and objects parts [37]. Remarkably, these concepts are acquired implicitly, without ever providing the network with information about the *location* of such structures in images. Hence, CNNs trained for image classification may already contain implicitly most of the information required to perform object detection.

We are not the first to address the problem of WSD with CNNs. The method of Wang *et al*. [36], for example, uses a pre-trained CNN to describe image regions and then learn object categories as corresponding visual topics. While this method is currently state-of-the-art in weakly supervised object detection, it comprises several components beyond the CNN and requires signifiant tuning.

In this paper we contribute a novel *end-to-end* method for weakly supervised object detection using pre-trained

CNNs which we call a *weakly supervised deep detection network* (WSDDN) (fig. 1). Our method (section 3) starts from an existing network, such as AlexNet pre-trained on ImageNet data, and extends it to reason explicitly and efficiently about image regions $R$. In order to do so, given an image $\mathbf{x}$, the first step is to efficiently extract region-level descriptors $\phi(\mathbf{x}; R)$ by inserting a spatial pyramid pooling layer on top of the convolutional layers of the CNN [14, 11]. Next, the network is branched to extract *two data streams* from the pooled region-level features. The first stream associates a class score $\phi^c(\mathbf{x}; R)$ to each region individually, performing *recognition*. The second stream, instead, *compares* regions by computing a probability distribution $\phi^d(\mathbf{x}; R)$ over them; the latter represents the belief that, among all the candidate regions in the image, $R$ is the one that contains the most salient image structure, and is therefore a proxy to *detection*. The recognition and detection scores computed for all the image regions are finally aggregated in order to predict the class of the image as a whole, which is then used to inject image-level supervision in learning.

It is interesting to compare our method to the most common weakly supervised object detection technique, namely multiple instance learning (MIL) [7]. MIL alternates between selecting which regions in images look like the object of interest and estimating an appearance model of the object using the selected regions. Hence, MIL uses the appearance model itself to perform region selection. Our technique differs from MIL in a fundamental way as regions are selected by a dedicated *parallel detection branch* in the network, which is independent of the recognition branch. In this manner, our approach helps avoiding one of the pitfalls of MIL, namely the tendency of the method to get stuck in local optima.

Our two-stream CNN is also weakly related to the recent work of Lin *et al*. [22]. They propose a "bilinear" architecture where the output of two parallel network streams are combined by taking the outer product of feature vectors at corresponding spatial locations. The authors state that this construction is inspired by the ventral and dorsal streams of the human visual system, one focusing on recognition and the other one on localisation. While our architecture contains two such streams, the similarity is only superficial. A key difference is that in Lin *et al*. the two streams are perfectly symmetric, and therefore there is no reason to believe that one should perform classification and the other detection; in our scheme, instead, the detection branch is explicitly designed to compare regions, breaking the symmetry. Note also that Lin *et al*. [22] do not perform WSD nor evaluate object detection performance.

Once the modifications have been applied, the network is ready to be fine-tuned on a target dataset, using only image-level labels, region proposals and back-propagation. In section 4 we show that, when fine-tuned on the PASCAL VOC training set, this architecture achieves state-of-the-art weakly supervised object detection on the PASCAL data, achieving superior results to the current state-of-the-art [36] but *using only CNN machinery*. Since the system can be trained end-to-end using standard CNN packages, it is also as efficient as the recent fully-supervised Fast R-CNN detector of Girshick *et al*. [11], both in training and in testing. Finally, as a byproduct of our construction we also obtain a powerful image classifier that *performs better than standard fine-tuning techniques* on the target data. Our findings are summarised in section 5.

## 2. Related Work

The majority of existing approaches to WSD formulate this task as MIL. In this formulation an image is interpreted as a bag of regions. If the image is labeled as positive, then one of the regions is assume to tightly contain the object of interest. If the image is labeled as negative, then no region contains the object. Learning alternates between estimating a model of the object appearance and selecting which regions in the positive bags correspond to the object using the appearance model.

The MIL strategy results in a non-convex optimization problem; in practice, solvers tend to get stuck in local optima such that the quality of the solution strongly depends on the initialization. Several papers have focused on developing various initialization strategies [19, 5, 32, 4] and on regularizing the optimization problem [31, 1]. Kumar *et al*. [19] propose a self-paced learning strategy that progressively includes harder samples to a small set of initial ones at training. Deselaers *et al*. [5] initialize object locations based on the objectness score. Cinbis *et al*. [4] propose a multi-fold split of the training data to escape local optima. Song *et al*. [31] apply Nesterov's smoothing technique [23] to the latent SVM formulation [10] to be more robust against poor initializations. Bilen *et al*. [1] propose a smoothed version of MIL that softly labels object instances instead of choosing the highest scoring ones. Additionally, their method regularizes the latent object locations by penalizing unlikely configurations based on symmetry and mutual exclusion principles.

Another line of research in WSD [31, 32, 36] is based on the idea of identifying the similarity between image parts. Song *et al*. [31] propose a discriminative graph-based algorithm that selects a subset of windows such that each window is connected to its nearest neighbors in positive images. In [32], the same authors extend this method to discover multiple co-occurring part configurations. Wang *et al*. [36] propose an iterative technique that applies a latent semantic clustering via latent Semantic Analysis (pLSA) on the windows of positive samples and selects the most discriminative cluster for each class based on its classification per-

formance. Bilen *et al*. [2] propose a formulation that jointly learns a discriminative model and enforces the similarity of the selected object regions via a discriminative convex clustering algorithm.

Recently a number of researchers [25, 26] have proposed weakly supervised localization principles to improve classification performance of CNNs without providing any annotation for the location of objects in images. Oquab *et al*. [25] employ a pre-trained CNN to compute a mid-level image representation for images of PASCAL VOC. In their follow-up work, Oquab *et al*. [26] modify a CNN architecture to *coarsely* localize object instances in image while predicting its label.

Jaderberg *et al*. [16] proposed a CNN architecture in which a subnetwork automatically pre-transforms an image in order to optimize the classification accuracy of a second subnetwork. This "transformer network", which is trained in an end-to-end fashion from image-level labels, is shown to align objects to a common reference frame, which is a proxy to detection. Our architecture contains a mechanism that pre-select image regions that are likely to contain the object, also trained in an end-to-end fashion; while this may seem very different, this mechanism can also be thought as learning transformations (as the ones that map the detected regions to a canonical reference frame). However, the nature of the selection process in in our and their networks are very different.

## 3. Method

In this section we introduce our *weakly supervised deep detection network* (WSDDN) method. The overall idea consists of three steps. First, we obtain a CNN pre-trained on a large-scale image classification task (section 3.1). Second, we construct the WSDDN as an architectural modification of this CNN (section 3.2). Third, we train/fine-tune the WSDDN on a target dataset, once more using only image-level annotations (section 3.3). The remainder of this section discusses these three steps in detail.

### 3.1. Pre-trained network

We build our method on a pre-trained CNN that has been pre-trained on the ImageNet ILSVRC 2012 data [28] with only image-level supervision (*i.e*. no bounding box annotations). We give the details of the used CNN architectures in section 4.

### 3.2. Weakly supervised deep detection network

Given the pre-trained CNN, we transform it into a WSDDN by introducing three modifications (see also section 3). First, we replace the last pooling layer immediately following the ReLU layer in the last convolutional block (also known as *relu5* and *pool5*, respectively) with a layer implementing *spatial pyramid pooling* (SPP) [20, 15]. This results in a function that takes as input an image $\mathbf{x}$ and a region (bounding box) $R$ and produces as output a feature vector or representation $\phi(\mathbf{x}; R)$. Importantly, the function decomposes as

$$\phi(\mathbf{x}; R) = \phi_{\text{SPP}}(\cdot; R) \circ \phi_{\text{relu5}}(\mathbf{x})$$

where $\phi_{\text{relu5}}(\mathbf{x})$ needs to be computed only once for the whole image and $\phi_{\text{SPP}}(\cdot; R)$ is fast to compute for any given region $R$. In practice, SPP is configured to be compatible to the first fully connected layers of networks (*i.e*. fc6). Note that SPP is implemented as a network layer as in [11] to allow to train the system end-to-end (and for efficiency).

Given an image $\mathbf{x}$, a shortlist of candidate object regions $\mathcal{R} = (R_1, \dots, R_n)$ are obtained by a region proposal mechanism. Here we experiment with two methods, Selective Search Windows (SSW) [34] and Edge Boxes (EB) [38]. As in [11], we then modify the SPP layer to take as input not a single region, but rather the full list $\mathcal{R}$; in particular, $\phi(\mathbf{x}; \mathcal{R})$ is defined as the concatenation of $\phi(\mathbf{x}; R_1), \dots, \phi(\mathbf{x}; R_n)$ along the fourth dimension (since each individual $\phi(\mathbf{x}; R)$ is a 3D tensor).

At this point in the architecture, region-level features are further processed by two fully connected layers $\phi_{\text{fc6}}$ and $\phi_{\text{fc7}}$, each comprising a linear map followed by a ReLU. Out of the output of the last such layer, we branch off two data streams, described next.

**Classification data stream.** The first data stream performs *classification* of the individual regions, by mapping each of them to a $C$-dimensional vector of class scores, assuming that the system is trained to detect $C$ different classes. This is achieved by evaluating a linear map $\phi_{fc8c}$ and results in a matrix of data $\mathbf{x}^c \in \mathbb{R}^{C \times |\mathcal{R}|}$, containing the class prediction scores for each region. The latter is then passed through a *softmax* operator, defined as follows:

$$[\sigma_{\text{class}}(\mathbf{x}^c)]_{ij} = \frac{e^{x^c_{ij}}}{\sum_{k=1}^{C} e^{x^c_{kj}}}. \tag{1}$$

**Detection data stream.** The second data stream performs instead *detection*, by scoring regions relative to one another. This is done on a class-specific basis by using a second linear map $\phi_{fc8d}$, also resulting in a matrix of scores $\mathbf{x}^d \in \mathbb{R}^{C \times |\mathcal{R}|}$. It is then passed through another softmax operator, but this time defined as follows:

$$[\sigma_{\text{det}}(\mathbf{x}^d)]_{ij} = \frac{e^{x^d_{ij}}}{\sum_{k=1}^{|\mathcal{R}|} e^{x^d_{ik}}}. \tag{2}$$

While the two streams are remarkably similar, the introduction of the $\sigma_{\text{class}}$ and $\sigma_{\text{det}}$ non-linearities in the classification and detection streams is a key difference which allows

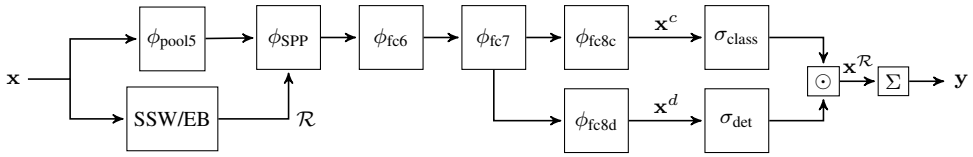


Figure 2. **Weakly-supervised deep detection network.** The figure illustrates the architecture of WSDDN.

to interpret them as performing classification and detection, respectively. In the first case, in fact, the softmax operator compares, for each region independently, class scores, whereas in the second case the softmax operator compares, for each class independently, the scores of different regions. Hence, the first branch predicts which class to associate to a region, whereas the second branch selects which regions are more likely to contain an informative image fragment.

**Combined region scores and detection.** The final score of each region is obtained by taking the element-wise (Hadamard) product $\mathbf{x}^{\mathcal{R}} = \sigma_{\text{class}}(\mathbf{x}^c) \odot \sigma_{\text{det}}(\mathbf{x}^d)$ of the two scoring matrices. The region scores are then used to rank image regions by likelihood of centring an object (for each class independently); standard non-maxima suppression is then performed (by iteratively removing regions with Intersection over Union (IoU) larger than 40% with regions already selected) to obtain the final list of class-specific detections in an image.

The way the two streams' scores are combined is reminiscent of the bilinear networks of [22], but there are three key differences. The first difference is that the introduction of the different softmax operators explicitly breaks the symmetry of the two streams. The second one is that, instead of computing the outer product of the two feature vectors $\sigma_{\text{class}}(\mathbf{x}^c_r) \otimes \sigma_{\text{det}}(\mathbf{x}^d_r)$, we compute the element-wise product $\sigma_{\text{class}}(\mathbf{x}^c_r) \odot \sigma_{\text{det}}(\mathbf{x}^d_r)$ (generating quadratically less parameters). The third difference is that scores $\sigma_{\text{class}}(\mathbf{x}^c_r) \otimes \sigma_{\text{det}}(\mathbf{x}^d_r)$ are computed for specific image regions $r$ rather than a fixed set of image locations on a grid. Together, these three differences mean that we can interpret $\sigma_{\text{det}}(\mathbf{x}^d)$ as a term that ranks regions, whereas $\sigma_{\text{class}}(\mathbf{x}^c)$ ranks classes. It is more difficult to clearly assess the nature of the two streams in [22].

**Image-level classification scores.** So far, WSDDN has computed region-level scores $\mathbf{x}^{\mathcal{R}}$. This is transformed in an image-level class prediction score by summation over regions:

$$y_c = \sum_{r=1}^{|\mathcal{R}|} x^{\mathcal{R}}_{cr}.$$

Note that both $y_c$ is a sum of element-wise product of softmax normalised scores over $|\mathcal{R}|$ regions and thus it is in the range of $(0, 1)$. Softmax is not performed at this stage as images are allowed to contain more than one object class (whereas regions should contain a single class).

### 3.3. Training WSDDN

Having discussed the WSDDN architecture in the previous section, here we explain how the model is trained. The data is a collection of images $\mathbf{x}_i, i = 1, \ldots, n$ with *image level labels* $\mathbf{y}_i \in \{-1, 1\}^C$. We denote by $\phi^{\mathbf{y}}(\mathbf{x}|\mathbf{w})$ the complete architecture, mapping an image $\mathbf{x}$ to a vector of class scores $\mathbf{y} \in \mathbb{R}^C$. The parameters $\mathbf{w}$ of the model lump together the coefficients of all the filters and biases in the convolutional and fully-connected layers. Then, stochastic gradient descent with momentum is used to optimise the energy function

$$E(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n}\sum_{k=1}^{C} \log(y_{ki}(\phi^{\mathbf{y}}_k(\mathbf{x}_i|\mathbf{w}) - \frac{1}{2}) + \frac{1}{2}), \tag{3}$$

hence optimising a sum of $C$ binary-log-loss terms, one per class. As $\phi^{\mathbf{y}}_k(\mathbf{x}_i|\mathbf{w})$ is in range of $(0, 1)$, it can be considered as a probability of class $k$ being present in image $\mathbf{x}_i$, *i.e.* $p(y_{ki} = 1)$. When the ground-truth label is positive, the binary log loss becomes $\log(p(y_{ki} = 1))$, $\log(1 - p(y_{ki} = 1))$ otherwise.

### 3.4. Spatial Regulariser

As WSDDN is optimised for image-level class labels, it does not guarantee any spatial smoothness such that if a region obtains a high score for an object class, the neighbouring regions with high overlap will also have high scores. In the supervised detection case, Fast-RCNN [11] takes the region proposals that have IoU with a ground truth box of at least $50\%$ as positive samples and learns to regress them into their corresponding ground truth bounding box. As our method does not have access to ground truth boxes, we follow a soft regularisation strategy that penalises the feature map discrepancies between the highest scoring region and the regions with at least $60\%$ IoU during training:

$$\frac{1}{nC}\sum_{k=1}^{C}\sum_{i=1}^{n^+_k} \frac{1}{2}\phi^{\mathbf{y}}_k(\mathbf{x}_i|\mathbf{w})(\phi^{\text{fc7}}_{kp} - \phi^{\text{fc7}}_{ki})^{\mathrm{T}}(\phi^{\text{fc7}}_{kp} - \phi^{\text{fc7}}_{ki})$$

where $n^+_k$ is the number of positive images for the class $k$ and $kp = \arg\max_j \phi^{\mathbf{y}}_{kj}$ is the highest scoring region in

image $i$ for the class $k$. We add this regularisation term to the cost function in eq. (3).

# 4. Experiments

In this section we conduct a thorough investigation of WSDDN and its components on weakly supervised detection and image classification.

## 4.1. Benchmark data.

We evaluate our method on the PASCAL VOC 2007 and 2010 datasets [9], as they are the most widely-used benchmark in weakly supervised object detection. While the VOC 2007 dataset consists of 2501 training, 2510 validation, and 5011 test images containing bounding box annotations for 20 object categories, VOC 2010 dataset contains 4998 training, 5105 validation, and 9637 test images for the same number of categories. We use the suggested training and validation splits and report results evaluated on *test* split. We report performance of our method on both the object detection and the image classification tasks of PASCAL VOC.

For detection, we use two performance measures. The first one follows the standard PASCAL VOC protocol and reports average precision (AP) at $50\%$ intersection-over-union (IoU) of the detected boxes with the ground truth ones. We also report CorLoc, a commonly-used weakly supervised detection measure [6]. CorLoc is the percentage of images that contain at least one instance of the target object class for which the most confident detected bounding box overlaps by at least $50\%$ with one of these instances. Differently from AP, which is measured on the PASCAL test set, CorLoc is evaluated on the union of the training and validation subset of PASCAL. For classification, we use the standard PASCAL VOC protocol and report AP.

## 4.2. Experimental setup.

We comprehensively evaluate our method with three pre-trained CNN models in our experiments as in [11]. The first network is the VGG-CNN-F [3] which is similar to AlexNet [18] but has reduced number of convolutional filters. We refer to this network as **S**, for small. The second one is VGG-CNN-M-1024 which has the same depth as **S** but has smaller stride in the first convolutional layer. We name this network **M** for medium. The last network is the deep VGG-VD16 model [30] and we call this network **L** for large. These models, which are pre-trained on the ImageNet ILSVRC 2012 challenge data [28], attain $18.8\%$, $16.1\%$ and $9.9\%$ top-5 accuracy respectively (using a single centre-crop) on ILSVRC (importantly no bounding box information is provided during pre-training). As explained in section 3.1, we apply the following modifications to the network. First, we replace the last pooling layer *pool5* with a SPP layer [15] which is configured to be compatible with the network's first fully connected layer. Second, we add a parallel detection branch to the classification one that contains a fully-connected layer followed by a soft-max layer. Third, we combine the classification and detection streams by element-wise product followed by summing scores across regions, and feed the latter to a binary log-loss layer. Note that this layer assesses the classification performance for the 20 classes together, but each of them is treated as a different binary classification problem; the reason is that classes can co-occur in the PASCAL VOC, such that the softmax log loss used in AlexNet is not appropriate.

The WSDDNs are trained on the PASCAL VOC training and validation data by using fine-tuning on all layers, a widely-adopted technique to improve the performance of a CNN on a target domain [3]. Here, however, fine tuning performs the essential function of learning the classification and detection streams, effectively causing the network to learn to detect objects, but using only weak image-level supervision. The experiments are run for $20$ epochs and all the layers are fine-tuned with the learning rate $10^{-5}$ for the first ten epochs and $10^{-6}$ for the last ten epochs. Each minibatch contains all region proposals from a single image.

In order to generate candidate regions to use with our networks, we evaluate two proposal methods, Selective Search Windows (SSW) [34] using its *fast* setting, and EdgeBoxes (EB) [38]. In addition to region proposals, EB provides an objectness score for each region based on the number of contours wholly encloses. We exploit this additional information by multiplying the feature map $\phi_{\text{SPP}}$ proportional to its score via a scaling layer in WSDDN and denote this setting as *Box Sc*. Since we use a SPP layer to aggregate descriptors for each region, images do not need to be resized to a particular size as in the original pre-trained model. Instead, we keep the original aspect ratio of images fixed and resize them to five different scales (setting their maximum of width or height to $\{480, 576, 688, 864, 1200\}$ respectively) as in [15]. During training, we apply random horizontal flips to the images and select a scale at random as a form of jittering or data augmentation. At test time we average the outputs of 10 images (*i.e*. the 5 scales and their flips). We use the publicly available CNN toolbox MatConvNet [35] to conduct our experiments and share our code, models and data [1].

When evaluated on an image, WSDDN produces, for each target class $c$ and image $\mathbf{x}$, a score $\mathbf{x}_r^{\mathcal{R}} = S_c(\mathbf{x}; r)$ for each region $r$ and an aggregated score $y_c = S_c(\mathbf{x})$ for each image. Non-maxima suppression (with $40$ % IoU threshold) is applied to the regions and then the scored regions and images are pooled together to compute detection AP and CorLoc.

[1] https://github.com/hbilen/WSDDN

| | S | M | L | Ens. |
|---|---|---|---|---|
| SSW | 31.1 | 30.9 | 24.3 | 33.3 |
| EB | 31.5 | 30.9 | 25.5 | 34.2 |
| EB + Box Sc. | 33.4 | 32.7 | 30.4 | 36.7 |
| EB + Box Sc. + Sp. Reg. | **34.5** | **34.9** | **34.8** | **39.3** |

Table 1. **VOC 2007 test** detection average precision (%). The ensemble network is denoted as **Ens**.

### 4.3. Detection results

**Baseline method.** First we design a single stream classification-detection network as an alternative baseline to WSDDN. Part of the construction is similar to WSDDN, as we replace *pool5* layer of VGG-CNN-F model with an SPP. However, we do not branch off two streams, but simply append to the last fully connected layer ($\phi_{\text{fc8c}}$) the following loss layer

$$\frac{1}{nC}\sum_{i=1}^{n}\sum_{k=1}^{C}\max\{0, 1 - y_{ki}\log\sum_{r=1}^{|\mathcal{R}|}\exp(x_{cr}^{\mathcal{R}})\}.$$

The term $\log\sum_{r=1}^{|\mathcal{R}|}\exp(x_{cr}^{\mathcal{R}})$ is a soft approximation of the max operator $\max_r x_{cr}^{\mathcal{R}}$ and was found to yield better performance than using the max scoring region. This observation is also reported in [1]. Note that the non-linearity is necessary as otherwise aggregating region-based scores would sum over the scores of a majority of regions that are uninformative. The loss function is once more a sum of $C$ binary hinge-losses, one for each class. This baseline obtains $21.6\%$ mAP detection score on the PASCAL VOC test set, which is well below the state-of-the-art ($31.6\%$ in [36]).

**Pre-trained CNN architectures.** We evaluate our method with the models **S**, **M** and **L** and also report the results for the ensemble of these models by simply averaging their scores. Table 1 shows that WSDDN with individual models **S** and **M** are already on par with the state-of-the-art method [36] and the ensemble outperforms the best previous score in the VOC 2007 dataset. Differently from supervised detection methods (*e.g.* [11]), detection performance of WSDDN does not improve with use of wider or deeper networks. In contrast, model **L** performs significantly worse than models **S** and **M** (see table 1). This can be explained with the fact that model **L** frequently focuses on parts of objects, instead of whole instances, and is still able to associate these parts with object categories due to its smaller convolution strides, higher resolution and deeper architecture.

**Object proposals.** Next, we compare the detection performances with two popular object proposal methods, SSW [34] and EB [38]. While both the region proposals provides comparable quality region proposals, using box scores of EB (denoted as *Box Sc* in table 1) leads to a $2\%$ improvement for models **S** and **M** and boosts the detection performance of model **L** $5\%$.

**Spatial regulariser.** We denote the setting where WSDDN is trained with the additional spatial regularisation term (denoted as *Sp. Reg.* in table 1). Finally the introduction of the regularisation improves the detection performance $1$, $2$ and $4$ mAP points for models **S**, **M** and **L** respectively. The improvements show that larger network benefits more from introduction of the spatial invariance around high confidence regions.

**Comparison with the state of the art.** After evaluating the design decisions, we follow the best setting (last row in table 1) and compare WSDDN to the state of the art in weakly supervised detection literature in table 2 and table 3 for the VOC 2007 dataset and in table 5 and table 6 for the VOC 2010 dataset. The results show that our method already achieves overall significantly better performance than these alternatives with a single model and ensemble models further boost the performance. The majority of previous work [31, 32, 1, 36, 2] use the Caffe reference CNN model [17], which is comparable to model **S** in this paper, as a black box to extract features over SSW proposals. In addition to CNN features, Cinbis *et al*. [4] use Fisher Vectors [27] and EB objectness measure of Zitnick and Dollar [38] as well. Differently from the previous work, WSDDN is based on a simple modification of the original CNN architecture fine-tuned on the target data using back-propagation.

Next, we investigate the results in more detail. While our method significantly outperforms the alternatives in majority of categories, is not as strong in chair, person and pottedplant categories. Failure and success case are illustrated in fig. 3. It can be noted that, by far, the most important failure modality for our system is that an object part (e.g. person face) is detected instead as the object as a whole. This can be explained by the fact that parts such as "face" are very often much more discriminative and with a less variable appearance than the rest of the object. Note that the root cause for this failure modality is that we, as many other authors, define objects as image regions that are most predictive for a given object class, and these may not include the object as a whole. Addressing this issue will therefore require incorporating additional cue in the model to try to learn the "whole object".

The output of our model could also be used as input to one of the existing methods for weakly-supervised detection that use a CNN as a black-box for feature extraction. Investigating this option is left to future work.

| method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | persn | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WSDDN **S** | 42.9 | 56.0 | 32.0 | 17.6 | 10.2 | 61.8 | 50.2 | 29.0 | 3.8 | 36.2 | 18.5 | 31.1 | **45.8** | 54.5 | 10.2 | 15.4 | 36.3 | 45.2 | 50.1 | 43.8 | 34.5 |
| WSDDN **M** | 43.6 | 50.4 | 32.2 | **26.0** | 9.8 | 58.5 | 50.4 | 30.9 | 7.9 | 36.1 | 18.2 | 31.7 | 41.4 | 52.6 | 8.8 | 14.0 | 37.8 | 46.9 | 53.4 | 47.9 | 34.9 |
| WSDDN **L** | 39.4 | 50.1 | 31.5 | 16.3 | 12.6 | 64.5 | 42.8 | **42.6** | 10.1 | 35.7 | 24.9 | 38.2 | 34.4 | 55.6 | 9.4 | 14.7 | 30.2 | 40.7 | 54.7 | 46.9 | 34.8 |
| WSDDN Ensemble | 46.4 | **58.3** | **35.5** | 25.9 | **14.0** | **66.7** | **53.0** | 39.2 | 8.9 | **41.8** | **26.6** | **38.6** | 44.7 | **59.0** | 10.8 | 17.3 | **40.7** | **49.6** | **56.9** | **50.8** | **39.3** |
| Bilen *et al*. [1] | 42.2 | 43.9 | 23.1 | 9.2 | 12.5 | 44.9 | 45.1 | 24.9 | 8.3 | 24.0 | 13.9 | 18.6 | 31.6 | 43.6 | 7.6 | **20.9** | 26.6 | 20.6 | 35.9 | 29.6 | 26.4 |
| Bilen *et al*. [2] | 46.2 | 46.9 | 24.1 | 16.4 | 12.2 | 42.2 | 47.1 | 35.2 | 7.8 | 28.3 | 12.7 | 21.5 | 30.1 | 42.4 | 7.8 | 20.0 | 26.8 | 20.8 | 35.8 | 29.6 | 27.7 |
| Cinbis *et al*. [4] | 39.3 | 43.0 | 28.8 | 20.4 | 8.0 | 45.5 | 47.9 | 22.1 | 8.4 | 33.5 | 23.6 | 29.2 | 38.5 | 47.9 | **20.3** | 20.0 | 35.8 | 30.8 | 41.0 | 20.1 | 30.2 |
| Wang *et al*. [36] | 48.8 | 41.0 | 23.6 | 12.1 | 11.1 | 42.7 | 40.9 | 35.5 | **11.1** | 36.6 | 18.4 | 35.3 | 34.8 | 51.3 | 17.2 | 17.4 | 26.8 | 32.8 | 35.1 | 45.6 | 30.9 |
| Wang *et al*. [36]+context | **48.9** | 42.3 | 26.1 | 11.3 | 11.9 | 41.3 | 40.9 | 34.7 | 10.8 | 34.7 | 18.8 | 34.4 | 35.4 | 52.7 | 19.1 | 17.4 | 35.9 | 33.3 | 34.8 | 46.5 | 31.6 |

Table 2. **VOC 2007 test** detection average precision (%). Comparison of our WSDDN on PASCAL VOC 2007 to the state-of-the-art in terms of AP.

| method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | persn | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WSDDN **S** | 68.5 | 67.5 | 56.7 | 34.3 | 32.8 | 69.9 | 75.0 | 45.7 | 17.1 | 68.1 | 30.5 | 40.6 | 67.2 | 82.9 | 28.8 | 43.7 | 71.9 | 62.0 | 62.8 | 58.2 | 54.2 |
| WSDDN **M** | 65.1 | 63.4 | 59.7 | 45.9 | 38.5 | 69.4 | 77.0 | 50.7 | 30.1 | **68.8** | 34.0 | 37.3 | 61.0 | 82.9 | 25.1 | 42.9 | **79.2** | 59.4 | **68.2** | 64.1 | 56.1 |
| WSDDN **L** | 65.1 | 58.8 | 58.5 | 33.1 | 39.8 | 68.3 | 60.2 | **59.6** | **34.8** | 64.5 | 30.5 | 43.0 | 56.8 | 82.4 | 25.5 | 41.6 | 61.5 | 55.9 | 65.9 | 63.7 | 53.5 |
| WSDDN Ensemble | 68.9 | **68.7** | **65.2** | 42.5 | **40.6** | **72.6** | 75.2 | 53.7 | 29.7 | 68.1 | 33.5 | 45.6 | 65.9 | **86.1** | 27.5 | **44.9** | 76.0 | **62.4** | 66.3 | **66.8** | **58.0** |
| Bilen *et al*. [2] | 66.4 | 59.3 | 42.7 | 20.4 | 21.3 | 63.4 | 74.3 | **59.6** | 21.1 | 58.2 | 14.0 | 38.5 | 49.5 | 60.0 | 19.8 | 39.2 | 41.7 | 30.1 | 50.2 | 44.1 | 43.7 |
| Cinbis *et al*. [4] | 65.3 | 55.0 | 52.4 | **48.3** | 18.2 | 66.4 | **77.8** | 35.6 | 26.5 | 67.0 | **46.9** | **48.4** | **70.5** | 69.1 | **35.2** | 35.2 | 69.6 | 43.4 | 64.6 | 43.7 | 52.0 |
| Wang *et al*. [36] | **80.1** | 63.9 | 51.5 | 14.9 | 21.0 | 55.7 | 74.2 | 43.5 | 26.2 | 53.4 | 16.3 | 56.7 | 58.3 | 69.5 | 14.1 | 38.3 | 58.8 | 47.2 | 49.1 | 60.9 | 48.5 |

Table 3. **VOC 2007 trainval** correct localization (CorLoc [6]) on positive *trainval* images (%).

| method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | persn | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WSDDN **S** | 92.5 | 89.9 | 89.5 | 88.3 | 66.5 | 83.6 | 92.1 | 90.3 | 73.0 | 85.7 | 72.6 | 91.4 | 90.1 | 89.0 | 94.4 | 78.1 | 86.0 | 76.1 | 91.1 | 85.5 | 85.3 |
| WSDDN **M** | 93.9 | 91.0 | 90.4 | 89.3 | 72.7 | 86.4 | 91.9 | 91.5 | 73.8 | 85.6 | 74.9 | 91.9 | 91.5 | 89.9 | 94.5 | 78.6 | 85.0 | 78.6 | 91.5 | 85.7 | 86.4 |
| WSDDN **L** | 93.3 | **93.9** | 91.6 | **90.8** | **82.5** | **91.4** | **92.9** | **93.0** | 78.1 | **90.5** | **82.3** | **95.4** | **92.7** | 92.4 | 95.1 | **83.4** | **90.5** | 80.1 | 94.5 | **89.6** | **89.7** |
| WSDDN Ensemble | 95.0 | 92.6 | 91.2 | 90.4 | 79.0 | 89.2 | 92.8 | 92.4 | **78.5** | **90.5** | 80.4 | 95.1 | 91.6 | **92.5** | 94.7 | 82.2 | 89.9 | **80.3** | 93.1 | 89.1 | 89.0 |
| Oquab *et al*. [24] | 88.5 | 81.5 | 87.9 | 82.0 | 47.5 | 75.5 | 90.1 | 87.2 | 61.6 | 75.7 | 67.3 | 85.5 | 83.5 | 80.0 | **95.6** | 60.8 | 76.8 | 58.0 | 90.4 | 77.9 | 77.7 |
| SPP [15] | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 82.4 |
| VGG-F [3] | 88.7 | 83.9 | 87.0 | 84.7 | 46.9 | 77.5 | 86.3 | 85.4 | 58.6 | 71.0 | 72.6 | 82.0 | 87.9 | 80.7 | 91.8 | 58.5 | 77.4 | 66.3 | 89.1 | 71.3 | 77.4 |
| VGG-M-1024 [3] | 91.4 | 86.9 | 89.3 | 85.8 | 53.3 | 79.8 | 87.8 | 88.6 | 59.0 | 77.2 | 73.1 | 85.9 | 88.3 | 83.5 | 91.8 | 59.9 | 81.4 | 68.3 | 93.0 | 74.1 | 79.9 |
| VGG-S [3] | **95.3** | 90.4 | **92.5** | 89.6 | 54.4 | 81.9 | 91.5 | 91.9 | 64.1 | 76.3 | 74.9 | 89.7 | 92.2 | 86.9 | 95.2 | 60.7 | 82.9 | 68.0 | **95.5** | 74.4 | 82.4 |
| VGG-VD16 [30] | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 89.3 |

Table 4. **VOC 2007 test** classification average precision (%).

| method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | persn | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WSDDN Ensemble | **57.4** | **51.8** | **41.2** | **16.4** | **22.8** | **57.3** | **41.8** | **34.8** | **13.1** | **37.6** | 10.8 | **37.0** | **45.2** | **64.9** | 14.1 | **22.3** | **33.8** | **27.6** | **49.1** | **44.8** | **36.2** |
| Cinbis *et al*. [4] | 44.6 | 42.3 | 25.5 | 14.1 | 11.0 | 44.1 | 36.3 | 23.2 | 12.2 | 26.1 | **14.0** | 29.2 | 36.0 | 54.3 | **20.7** | 12.4 | 26.5 | 20.3 | 31.2 | 23.7 | 27.4 |

Table 5. **VOC 2010 test** detection average precision (%). `http://host.robots.ox.ac.uk:8080/anonymous/3QGEGM.html`

| method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | persn | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WSDDN Ensemble | **77.4** | **73.2** | **61.9** | 39.6 | **50.8** | **84.4** | 67.5 | **49.6** | 38.6 | **73.4** | 30.4 | **53.2** | 72.9 | **84.1** | 30.3 | **53.1** | **76.6** | 48.5 | 61.6 | **66.7** | **59.7** |
| Cinbis *et al*. [4] | 61.1 | 65.0 | 59.2 | **44.3** | 28.3 | 80.6 | **69.7** | 31.2 | **42.8** | 73.3 | **38.3** | 50.2 | **74.9** | 70.9 | **37.3** | 37.1 | 65.3 | **55.3** | **61.7** | 58.2 | 55.2 |

Table 6. **VOC 2010 trainval** correct localization (CorLoc [6]) on positive *trainval* images (%).

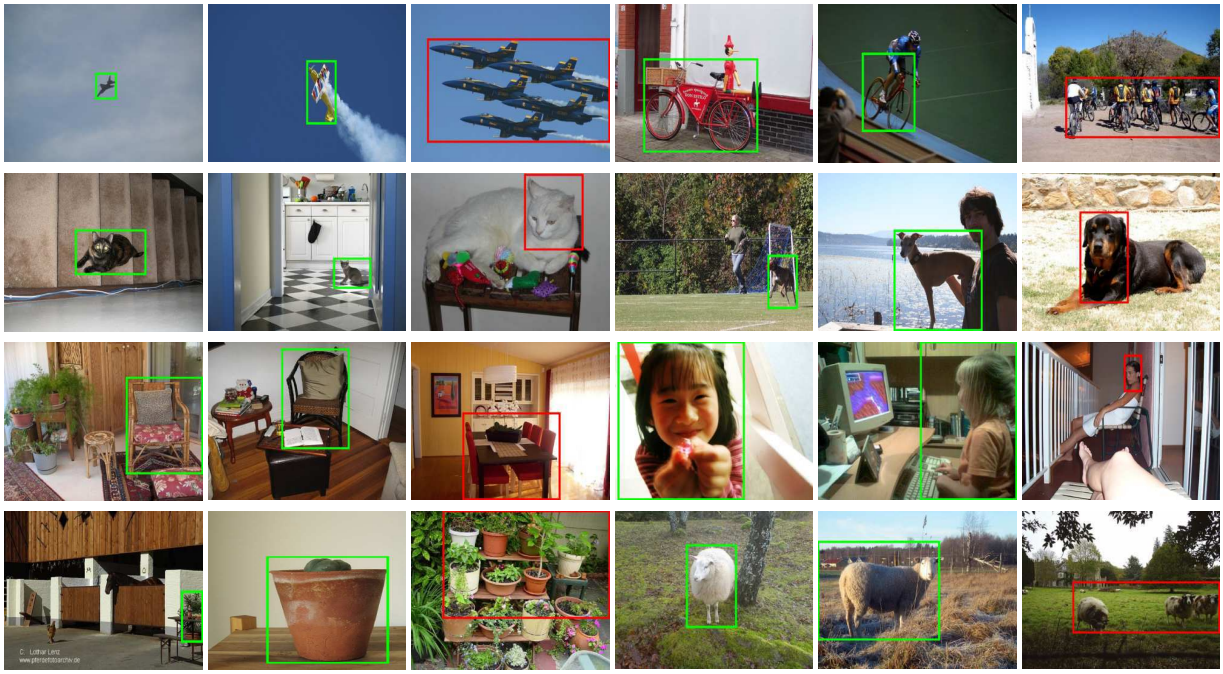

Figure 3. This figure depicts success (in green) and failure cases (in red) of our detector in randomly picked images. Majority of false detections contains two kinds of error: i) group multiple object instances with a single bounding box, ii) focus on (discriminative) parts (*e.g.* "faces") rather than whole object.

### 4.4. Classification Results

While WSDDN is primarily designed for weakly-supervised object detection, ultimately it is trained to perform image classification. Hence, it is interesting to evaluate its performance on this task as well. To this end, we use the PASCAL VOC 2007 benchmark and contrast it to standard fine-tuning techniques that are often used in combination with CNNs and show the results in table 6. These techniques have been thoroughly investigated in [3, 15, 24]. Chatfield *et al*. [3], in particular, analyse many variants of fine-tuning, including extensive data augmentation, on the PASCAL VOC. They experiment with three architectures, VGG-F, VGG-M, and VGG-S. While VGG-F is their fastest model, the other two networks are slower but more accurate. As explained in 4.2, we initialise WSDDN **S** and **M** with the pre-trained VGG-F and VGG-M-1024 respectively and thus they should be considered as right baselines. WSDDN **S** and **M** improves $8$ and $7$ points over VGG-F and VGG-M-1024 respectively.

We also compare WSDDN to the SPP-net [15] which uses the Overfeat-7 [29] with a 4-level spatial pyramid pooling layer $\{6 \times 6, 3 \times 3, 2 \times 2, 1 \times 1\}$ for supervised object detection. While they do not perform fine-tuning, they include a spatial pooling layer. Applied to image classification, their best performance on the PASCAL VOC 2007 is $82.4\%$. Finally we compare WSDDN **L** to the competitive VGG-VD16 [30]. Interestingly, this method also exploits coarse local information by aggregating the activations of the last fully connected layer over multiple locations and scales. WSDDN **L** outperforms this very competitive baseline with a margin of $0.4$ point.

## 5. Conclusions

In this paper, we have presented WSDDN, a simple modification of a pre-trained CNN for image classification that allows it to perform weakly supervised detection. It achieves significantly better performance than existing methods on weakly supervised detection, while requiring only fine-tuning on a target dataset using back-propagation, region proposals and image-level labels. Since it works on top of a SPP layer, it is also efficient at training and test time. WSDDN is also shown to perform better than traditional fine-tuning techniques to improve the performance of a pre-trained CNN on the problem of image classification.

We have identified the detection of object parts as a failure modality of the method, damaging its performance in selected object categories, and imputed that to the main criterion used to identify objects, namely the selection of highly-distinctive image regions. We are currently exploring complementary cues that would favour detecting complete objects instead.

**Acknowledgments:** This work acknowledges the support of the EPSRC grant EP/L024683/1 and the ERC Starting Grant IDIU.

## References


[1] H. Bilen, M. Pedersoli, and T. Tuytelaars. Weakly supervised object detection with posterior regularization. In *BMVC*, 2014.

[2] H. Bilen, M. Pedersoli, and T. Tuytelaars. Weakly supervised object detection with convex clustering. In *CVPR*, 2015.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[4] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *arXiv preprint arXiv:1503.00949*, 2015.

[5] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, pages 452–466. 2010.

[6] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *IJCV*, 100(3):275–293, 2012.

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

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[9] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[11] R. Girshick. Fast r-cnn. In *ICCV*, 2015.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312. 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pages 346–361. 2014.

[16] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *NIPS 2015*, 2015.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *International Conference on Multimedia*, pages 675–678, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105. 2012.

[19] M. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, pages 1189–1197, 2010.

[20] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[22] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 2015.

[23] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[25] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Weakly supervised object recognition with convolutional neural networks. In *CVPR*, 2014.

[26] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free?–weakly-supervised learning with convolutional neural networks. In *CVPR*, pages 685–694, 2015.

[27] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010.

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, April 2015.

[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR2014*, 2014.

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

[31] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell. On learning to localize objects with minimal supervision. In *ICML*, pages 1611–1619, 2014.

[32] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell. Weakly-supervised discovery of visual pattern configurations. In *NIPS*, pages 1637–1645, 2014.

[33] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014.

[34] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[35] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[36] C. Wang, W. Ren, K. Huang, and T. Tan. Weakly supervised object localization with latent category learning. In *ECCV 2014*, volume 8694, pages 431–445, 2014.

[37] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene CNNs. In *ICLR*, 2015.

[38] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405. 2014.