Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks

Maxime Oquab[1,*] Leon Bottou[2] Ivan Laptev[1,*] Josef Sivic[1,*]

[1]INRIA, Paris, France [2]MSR, New York, USA

Abstract

Convolutional neural networks (CNN) have recently shown outstanding image classification performance in the large-scale visual recognition challenge (ILSVRC2012). The success of CNNs is attributed to their ability to learn rich mid-level image representations as opposed to hand-designed low-level features used in other image classification methods. Learning CNNs, however, amounts to estimating millions of parameters and requires a very large number of annotated image samples. This property currently prevents application of CNNs to problems with limited training data.

In this work we show how image representations learned with CNNs on large-scale annotated datasets can be efficiently transferred to other visual recognition tasks with limited amount of training data. We design a method to reuse layers trained on the ImageNet dataset to compute mid-level image representation for images in the PASCAL VOC dataset. We show that despite differences in image statistics and tasks in the two datasets, the transferred representation leads to significantly improved results for object and action classification, outperforming the current state of the art on Pascal VOC 2007 and 2012 datasets. We also show promising results for object and action localization.

1. Introduction

Object recognition has been a driving motivation for research in computer vision for many years. Recent progress in the field has allowed recognition to scale up from a few object instances in controlled setups towards hundreds of object categories in arbitrary environments. Much of this progress has been enabled by the development of robust image descriptors such as SIFT [33] and HOG [8], bag-of-features image representations [7, 27, 37, 46] as well as deformable part models [14]. Another enabling factor has been the development of increasingly large and realistic image datasets providing object annotation for training and testing, such as Caltech256 [18], Pascal VOC [11] and ImageNet [9].

Although being less common in recent years, neural networks have a long history in visual recognition. Rosenblatt's Mark I Perceptron [40] arguably was one of the first computer vision systems. Inspired by the neural connectivity pattern discovered by Hubel and Wiesel [20], Fukushima's Neocognitron [16] extended earlier networks with invariance to image translations. Combining the back-propagation algorithm [41] with the Neocognitron architecture, convolutional neural networks [26, 30] quickly achieved excellent results in optical character recognition leading to large-scale industrial applications [31, 44].

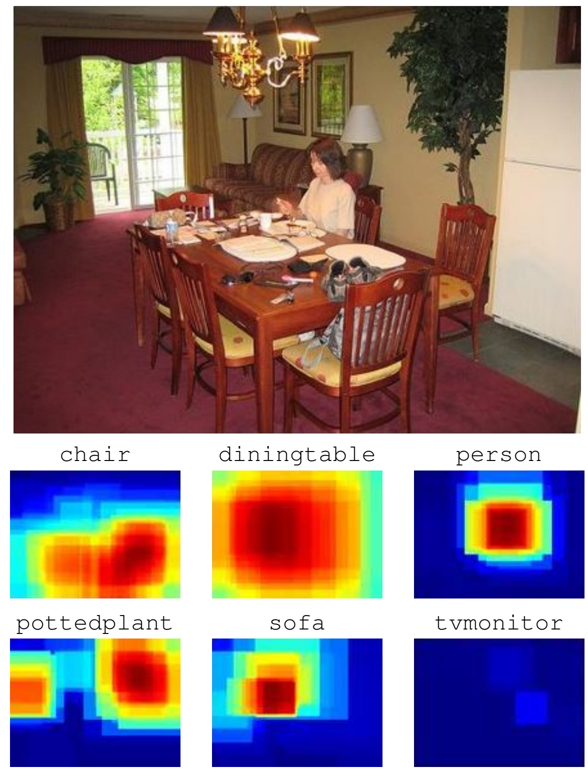

Figure 1: Recognition and localization results of our method for a Pascal VOC test image. Output maps are shown for six object categories with the highest responses.

Convolutional neural networks (CNN) are high-capacity classifiers with very large numbers of parameters that must be learned from training examples. While CNNs have been advocated beyond character recognition for other vision tasks [35, 51] including generic object recognition [32],

*WILLOW project-team, Département d'Informatique de l'École Normale Supérieure, ENS/Inria/CNRS UMR 8548, Paris, France.

their performance was limited by the relatively small sizes of standard object recognition datasets.

Notably, many successful image classification pipelines share aspects of the Neocognitron and convolutional neural networks. Quantizing and spatially aggregating local descriptors [7, 27, 33] arguably produces low-level image features comparable to those computed by the first two layers of the Neocognitron. It is therefore possible that these manually designed pipelines only outperformed earlier CNNs because CNNs are hard to train using small datasets.

This situation has changed with the appearance of the large-scale ImageNet dataset [9] and the rise of GPU computing. Krizhevsky *et al.* [25] achieve a performance leap in image classification on the ImageNet 2012 Large-Scale Visual Recognition Challenge (ILSVRC-2012), and further improve the performance by training a network on all 15 million images and 22,000 ImageNet classes. As much as this result is promising and exciting, it is also worrysome. Will we need to collect millions of annotated images for each new visual recognition task in the future?

It has been argued that computer vision datasets have significant differences in image statistics [50]. For example, while objects are typically centered in Caltech256 and ImageNet datasets, other datasets such as Pascal VOC and LabelMe are more likely to contain objects embedded in a scene (see Figure 3). Differences in viewpoints, scene context, "background" (negative class) and other factors, inevitably affect recognition performance when training and testing across different domains [38, 42, 50]. Similar phenomena have been observed in other areas such as NLP [22]. Given the "data-hungry" nature of CNNs and the difficulty of collecting large-scale image datasets, the applicability of CNNs to tasks with limited amount of training data appears as an important open problem.

To address this problem, we propose to transfer image representations learned with CNNs on large datasets to other visual recognition tasks with limited training data. In particular, we design a method that uses ImageNet-trained layers of CNN to compute efficient mid-level image representation for images in Pascal VOC. We analyze the transfer performance and show significant improvements on the Pascal VOC object and action classification tasks, outperforming the state of the art. We also show promising results for object and action localization. Results of object recognition and localization by our method are illustrated in Figure 1.

In the following we discuss related work in Section 2. Sections 3 and 4 present our method and experiments, respectively.

2. Related Work

Our method is related to numerous works on transfer learning, image classification, and deep learning, which we briefly discuss below.

Transfer learning. Transfer learning aims to transfer knowledge between related *source* and *target* domains [36]. In computer vision, examples of transfer learning include [4, 49] which try to overcome the deficit of training samples for some categories by adapting classifiers trained for other categories. Other methods aim to cope with different data distributions in the source and target domains for the same categories, e.g. due to lighting, background and view-point variations [13, 24, 42]. These and other related methods adapt classifiers or kernels while using standard image features. Differently to this work, we here transfer image representations trained on the source task.

More similar to our work, [3] trains CNNs on unsupervised pseudo-tasks. Differently to [3] we pre-train the convolutional layers of CNNs on a large-scale supervised task and address variations in scale and position of objects in the image. Transfer learning with CNNs has been also explored for Natural Language Processing [6] in a manner closely related to our approach. Other recent efforts done in parallel with our work also propose transferring image representations learnt from the large-scale fully-labelled ImageNet dataset using the convolutional neural network architecture of [25]. However, they investigate transfer to other visual recognition tasks such as Caltech256 image classification [53], scene classification [10] and object localization [17, 43].

Visual object classification. Most of the recent image classification methods follow the bag-of-features pipeline [7]. Densely-sampled SIFT descriptors [33] are typically quantized using unsupervised clustering (k-means, GMM). Histogram encoding [7, 46], spatial pooling [27] and more recent Fisher Vector encoding [37] are common methods for feature aggregation. While such representations have been shown to work well in practice, it is unclear whether they should be optimal for the task. This question raised considerable interest in the subject of mid-level features [5, 23, 45], and feature learning in general [29, 39, 48]. The goal of this work is to show that convolutional network layers provide generic mid-level image representations that can be transferred to new tasks.

Deep Learning. The recent revival of interest in multilayer neural networks was triggered by a growing number of works on learning intermediate representations, either using unsupervised methods, as in [19, 28], or using more traditional supervised techniques, as in [12, 25].

3. Transferring CNN weights

The CNN architecture of [25] contains more than 60 million parameters. Directly learning so many parameters from only a few thousand training images is problematic. The key idea of this work is that the internal layers of the CNN can act as a *generic extractor of mid-level image representation*, which can be pre-trained on one dataset (the *source task*, here ImageNet) and then re-used on other *target tasks*

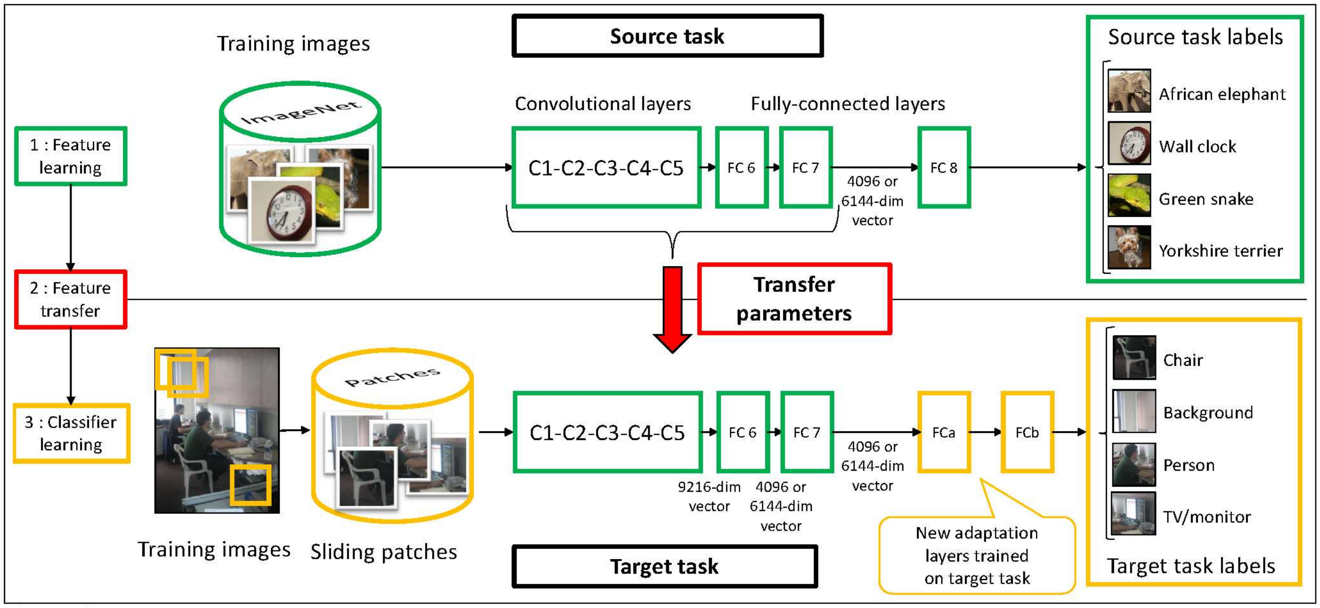

Figure 2: **Transferring parameters of a CNN.** First, the network is trained on the source task (ImageNet classification, top row) with a large amount of available labelled images. Pre-trained parameters of the internal layers of the network (C1-FC7) are then transferred to the target tasks (Pascal VOC object or action classification, bottom row). To compensate for the different image statistics (type of objects, typical viewpoints, imaging conditions) of the source and target data we add an adaptation layer (fully connected layers FCa and FCb) and train them on the labelled data of the target task.

(here object and action classification in Pascal VOC), as illustrated in Figure 2. However, this is difficult as the labels and the distribution of images (type of objects, typical viewpoints, imaging conditions, etc.) in the source and target datasets can be very different, as illustrated in Figure 3. To address these challenges we (i) design an architecture that explicitly remaps the class labels between the source and target tasks (Section 3.1), and (ii) develop training and test procedures, inspired by sliding window detectors, that explicitly deal with different distributions of object sizes, locations and scene clutter in source and target tasks (Sections 3.2 and 3.3).

3.1. Network architecture

For the source task, we use the network architecture of Krizhevsky *et al.* [25]. The network takes as input a square 224×224 pixel RGB image and produces a distribution over the ImageNet object classes. This network is composed of five successive convolutional layers C1...C5 followed by three fully connected layers FC6...FC8 (Figure 2, top). Please refer to [25] for the description of the geometry of the five convolutional layers and their setup regarding contrast normalization and pooling. The three fully connected layers then compute $\mathbf{Y}_6 = \sigma(\mathbf{W}_6\mathbf{Y}_5 + \mathbf{B}_6)$, $\mathbf{Y}_7 = \sigma(\mathbf{W}_7\mathbf{Y}_6 + \mathbf{B}_7)$, and $\mathbf{Y}_8 = \psi(\mathbf{W}_8\mathbf{Y}_7 + \mathbf{B}_8)$, where $\mathbf{Y}_k$ denotes the output of the k-th layer, $\mathbf{W}_k$, $\mathbf{B}_k$ are the trainable parameters of the k-th layer, and $\sigma(\mathbf{X})[i] = \max(0, \mathbf{X}[i])$ and $\psi(\mathbf{X})[i] = e^{\mathbf{X}[i]} / \sum_j e^{\mathbf{X}[j]}$ are the "ReLU" and "SoftMax" non-linear activation functions.

For target tasks (Pascal VOC object and action classification) we wish to design a network that will output scores for target categories, or `background` if none of the categories are present in the image. However, the object labels in the source task can be very different from the labels in the target task (also called a "label bias" [50]). For example, the source network is trained to recognize different breeds of dogs such as `husky dog` or `australian terrier`, but the target task contains only one label `dog`. The problem becomes even more evident for the target task of action classification. What object categories in ImageNet are related to the target actions `reading` or `running`?

In order to achieve the transfer, we remove the output layer FC8 of the pre-trained network and add an adaptation layer formed by two fully connected layers FCa and FCb (see Figure 2, bottom) that use the output vector $\mathbf{Y}_7$ of the layer FC7 as input. Note that $\mathbf{Y}_7$ is obtained as a complex non-linear function of potentially all input pixels and may capture mid-level object parts as well as their high-level configurations [28, 54]. The FCa and FCb layers compute $\mathbf{Y}_a = \sigma(\mathbf{W}_a\mathbf{Y}_7 + \mathbf{B}_a)$ and $\mathbf{Y}_b = \psi(\mathbf{W}_b\mathbf{Y}_a + \mathbf{B}_b)$, where $\mathbf{W}_a$, $\mathbf{B}_a$, $\mathbf{W}_b$, $\mathbf{B}_b$ are the trainable parameters. In all our experiments, FC6 and FC7 have equal sizes (either 4096 or 6144, see Section 4), FCa has size 2048, and FCb has a size equal to the number of target categories.

The parameters of layers C1...C5, FC6 and FC7 are first trained on the source task, then transferred to the target task and kept fixed. Only the adaptation layer is trained on the target task training data as described next.

Figure 3: Illustration of different dataset statistics between the source (ImageNet) and target (Pascal VOC) tasks. Pascal VOC data displays objects embedded in complex scenes, at various scales (right), and in complex mutual configurations (middle). Left: Image from ImageNet with label `maltese terrier`. Middle and right: Images from Pascal VOC with label `dog`.

3.2. Network training

First, we pre-train the network using the code of [25] on the ImageNet classification source task. Each image typically contains one object centered and occupying significant portion of the image with limited background clutter as illustrated in Figure 3(left). The network is trained to predict the ImageNet object class label given the entire image as input. Details are given in Section 4.

As discussed above, the network is pre-trained to classify source task images that depict single centered objects. The images in the target task, however, often depict complex scenes with multiple objects at different scales and orientations with significant amount of background clutter, as illustrated in Figure 3 (middle and right). In other words, the distribution of object orientations and sizes as well as, for example, their mutual occlusion patterns is very different between the two tasks. This issue has been also called "a dataset capture bias" [50]. In addition, the target task may contain many other objects in the background that are not present in the source task training data (a "negative data bias" [50]). To explicitly address these issues we train the adaptation layer using a procedure inspired by training sliding window object detectors (e.g. [15]) described next.

We employ a sliding window strategy and extract around 500 square patches from each image by sampling eight different scales on a regularly-spaced grid with at least 50% overlap between neighboring patches. More precisely, we use square patches of width $s = \min(w, h)/\lambda$ pixels, where w and h are the width and height of the image, respectively, and $\lambda \in \{1{,}1.3{,}1.6{,}2{,}2.4{,}2.8{,}3.2{,}3.6{,}4\}$. Each patch is rescaled to 224×224 pixels to form a valid input for the network.

Sampled image patches may contain one or more objects, background, or only a part of the object. To label patches in training images, we measure the overlap between the bounding box of a patch P and ground truth bounding boxes B of annotated objects in the image. The patch is labelled as a positive training example for class o if there exists a B_o corresponding to class o such that (i) B_o overlaps sufficiently with the patch $|P \cap B_o| \geq 0.2|P|$, (ii) the patch contains large portion of the object $|P \cap B_o| \geq 0.6|B_o|$, and (iii) the patch overlaps with no more than one object. In the above definitions $|A|$ measures the area of the bounding box A. Our labeling criteria are illustrated in Figure 4.

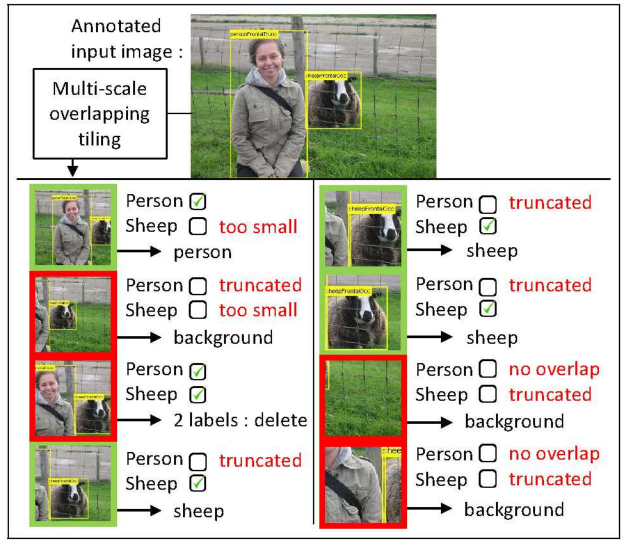

Figure 4: **Generating training data for the target task.** The input image (top) is divided into multi-scale overlapping patches (bottom). Each patch is labelled with an object label (green) or as background (red) depending on the overlap with object bounding boxes. Note that object patches are similar in appearance to the training data for the source task containing mostly centered objects.

Dealing with background. As discussed above, the target task has an additional `background` label for patches that do not contain any object. One additional difficulty is that the training data is unbalanced: most patches from training images come from background. This can be addressed by re-weighting the training cost function, which would amount to re-weighting its gradients during training. We opt for a slightly different procedure and instead re-sample the training patches to balance the training data distribution. This resampled training set is then used to form mini-batches for the stochastic gradient descent training. This is implemented by sampling a random 10% of the training background patches.

3.3. Classification

At test time we apply the network to each of the (approximately) 500 overlapping multi-scale patches extracted from the test image. Examples of patch scores visualized over entire images are shown in Figures 1 and 5. We use the following aggregation formula to compute the overall score for object C_n in the image

$$\text{score}(C_n) = \frac{1}{M} \sum_{i=1}^{M} y(C_n|P_i)^k, \tag{1}$$

where $y(C_n|P_i)$ is the output of the network for class C_n on image patch P_i, M is the number of patches in the image, and $k \geq 1$ is a parameter. Higher values of k focus on the highest scoring patches and attenuate the contributions of low- and mid-scoring patches. The value of $k = 5$ was optimized on the validation set and is fixed in our experiments.

Note that patch scores could be computed much more efficiently by performing large convolutions on adequately subsampled versions of the full image, as described for instance in [12]. This would permit a denser patch coverage at a lower computation cost.

4. Experiments

In this section we first describe details of training, and discuss pre-training results for the source task of ImageNet object classification. We next show experimental results of the proposed transfer learning method on the target Pascal VOC object classification task for both VOC 2007 and VOC 2012 datasets. We also investigate the dependency of results on the overlap of source and target tasks by object classes. Finally, we apply the proposed transfer learning method on a very different task of action recognition in still images.

Training convolutional networks. All our training sessions were carried out using the code provided by Krizhevsky *et al.* [25] and replicating their exact dropout and jittering strategies. However, we do not alter the RGB intensities and we use a single GeForce GTX Titan GPU with 6GB of memory instead of the two GPUs of earlier generation used in [25]. The training procedure periodically evaluates the cross-entropy objective function on a subset of the training set and on a validation set. The initial learning rates are set to 0.01 and the network is trained until the training cross-entropy is stabilized. The learning rates are then divided by 10 and the training procedure repeats. We stop training after three iterations. We have not tuned parameters for this part of the algorithm and we did not observe overfitting on the validation set.

Image classification on ImageNet. We first train a single convolutional network on the 1000 classes and 1.2 million images of the ImageNet 2012 Large Scale Visual Recognition Challenge (ILSVRC-2012). This network has exactly the same structure as the network described in [25]. Layers FC6 and FC7 have 4096 units. Training lasts about one week. The resulting network achieves a 18% top-5 error rate[1], comparable to the 17% reported by [25] for a single network. This slight performace loss could be caused by the absence of RGB intensity manipulation in our experiments.

Image classification on Pascal VOC 2007. We apply our mid-level feature transfer scheme to the Pascal VOC 2007 object classification task. Results are reported in Table 1. Our transfer technique (PRE-1000C) demonstrates significant improvements over previous results on this data outperforming the 2007 challenge winners [34] (INRIA) by 18.3% and the more recent work of [47] (NUS-PSL) by 7.2%.

[1]5 guesses are allowed.

Image classification on Pascal VOC 2012. We next apply our method to the Pascal VOC 2012 object classification task. Results are shown in the row PRE-1000C of Table 2. Although these results are on average about 4% inferior to those reported by the winners of the 2012 challenge (NUS-PSL [52]), our method outperforms [52] on five out of twenty classes. To estimate the performance boost provided by the feature transfer, we compare these results to the performance of an identical network directly trained on the Pascal VOC 2012 training data (NO PRETRAIN) without using any external data from ImageNet. Notably, the performance drop of nearly 8% in the case of NO PRETRAIN clearly indicates the positive effect of the proposed transfer.

Transfer learning and source/target class overlap. Our source ILSVRC-2012 dataset contains target-related object classes, in particular, 59 species of birds and 120 breeds of dogs related to the `bird` and `dog` classes of Pascal VOC. To understand the influence of this overlap on our results, we have pre-trained the network on a source task data formed by 1,000 ImageNet classes selected, this time, *at random* among all the 22,000 available ImageNet classes. Results of this experiment are reported in Table 2, row PRE-1000R. The overall performance has decreased slightly, indicating that the overlap between classes in the source and target domains may have a positive effect on the transfer. Given the relatively small performance drop, however, we conclude that our transfer procedure is robust to changes of source and target classes. As the number of training images in this experiment was about 25% smaller than in the ILSVRC-2012 training set (PRE-1000C), this could have been another reason for the decrease of performance.

Conversely, we have augmented the 1,000 classes of the ILSVRC-2012 training set with 512 additional ImageNet classes selected to increase the overlap with specific classes in the Pascal VOC target task. We included all the ImageNet classes located below the `hoofed mammal` (276 classes), `furniture` (165), `motor vehicle` (48), `public transport` (18), `bicycle` (5) nodes of the WordNet hierarchy. In order to accommodate the larger number of classes, we also increased the size of the FC6 and FC7 layers from 4,096 to 6,144 dimensions. Training on the resulting 1.6 million images achieves a 21.8% top-5 error rate on the 1,512 classes. Using this pre-trained network we have obtained further improvements on the target task, outperforming the winner of Pascal VOC 2012 [52] on average (row PRE-1512 in Table 2). In particular, improvements are obtained for categories (`cow`, `horse`, `sheep`, `sofa`, `chair`, `table`) related to the added classes in the source task. By comparing results for PRE-1000R, PRE-1000C and PRE-1512 setups, we also note the consistent improvement of *all* target classes. This suggests that the number of images and classes in the source task might be decisive for the performance in the target task. Hence, we expect further improvements by our method using larger source tasks.

	plane	bike	bird	boat	btl	bus	car	cat	chair	cow	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	**mAP**
INRIA [34]	77.5	63.6	56.1	71.9	33.1	60.6	78.0	58.8	53.5	42.6	54.9	45.8	77.5	64.0	85.9	36.3	44.7	50.6	79.2	53.2	59.4
NUS-PSL [47]	82.5	79.6	64.8	73.4	**54.2**	75.0	77.5	79.2	46.2	62.7	41.4	74.6	**85.0**	76.8	91.1	53.9	61.0	**67.5**	83.6	70.6	70.5
PRE-1000C	**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**

Table 1: Per-class results for object classification on the VOC2007 test set (average precision %).

	plane	bike	bird	boat	btl	bus	car	cat	chair	cow	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	**mAP**
NUS-PSL [52]	**97.3**	**84.2**	80.8	**85.3**	**60.8**	**89.9**	**86.8**	89.3	**75.4**	77.8	**75.1**	83.0	87.5	**90.1**	95.0	57.8	79.2	**73.4**	**94.5**	**80.7**	82.2
NO PRETRAIN	85.2	75.0	69.4	66.2	48.8	82.1	79.5	79.8	62.4	61.9	49.8	75.9	71.4	82.7	93.1	59.1	69.7	49.3	80.0	76.7	70.9
PRE-1000C	93.5	78.4	*87.7*	80.9	57.3	85.0	81.6	*89.4*	66.9	73.8	62.0	*89.5*	83.2	87.6	*95.8*	*61.4*	79.0	54.3	88.0	78.3	78.7
PRE-1000R	93.2	77.9	83.8	80.0	55.8	82.7	79.0	84.3	66.2	71.7	59.5	83.4	81.4	84.8	95.2	59.8	74.9	52.9	83.8	75.7	76.3
PRE-1512	94.6	82.9	**88.2**	84.1	60.3	89.0	84.4	**90.7**	72.1	**86.8**	69.0	**92.1**	**93.4**	88.6	**96.1**	**64.3**	**86.6**	62.3	91.1	79.8	**82.8**

Table 2: Per-class results for object classification on the VOC2012 test set (average precision %).

Action	jump	phon	instr	read	bike	horse	run	phot	comp	walk	**mAP**
STANFORD [1]	75.7	44.8	66.6	44.4	93.2	94.2	87.6	38.4	**70.6**	**75.6**	69.1
OXFORD [1]	**77.0**	**50.4**	65.3	39.5	**94.1**	**95.9**	**87.7**	42.7	68.6	74.5	69.6
NO PRETRAIN	43.2	30.6	50.2	25.0	76.8	80.7	75.2	22.2	37.9	55.6	49.7
PRE-1512	73.4	44.8	**74.8**	43.2	92.1	94.3	83.4	**45.7**	65.5	66.8	68.4
PRE-1512U	74.8	46.0	**75.6**	**45.3**	93.5	95.0	86.5	**49.3**	66.7	69.5	**70.2**

Table 3: Pascal VOC 2012 action classification results (AP %).

Varying the number of adaptation layers. We have also tried to change the number of adaptation layers in the best performing PRE-1512 training set-up. Using only one fully connected adaptation layer FCb of size 21 (the number of categories) results in about 1% drop in performance. Similarly, increasing the number of adaptation layers to three (of sizes 2048, 2048 and 21, respectively) also results in about 1% drop in classification performance.

Object localization. Although our method has not been explicitly designed for the task of localization, we have observed strong evidence of object and action localization provided by the network at test time. For qualitative assessment of localization results, we compute an output map for each category by averaging the scores of all the testing patches covering a given pixel of the test image. Examples of such output maps are given in Figures 1 and 5 as well as on the project webpage [2]. This visualization clearly demonstrates that the system knows the size and locations of target objects within the image. Addressing the detection task seems within reach.

Action recognition. The Pascal VOC 2012 action recognition task consists of 4588 training images and 4569 test images featuring people performing actions among ten categories such as `jumping`, `phoning`, `playing instrument` or `reading`. This fine-grained task differs from the object classification task because it entails recognizing fine differences in human poses (e.g. `running` v.s. `walking`) and subtle interactions with objects (`phoning` or `taking photo`). Training samples with multiple simultaneous actions are excluded from our training set.

To evaluate how our transfer method performs on this very different target task, we use a network pre-trained on 1512 ImageNet object classes and apply our transfer methodology to the Pascal VOC action classification task. Since the bounding box of the person performing the action is known at testing time, both training and testing are performed using a single square patch per sample, centered on the person bounding box. Extracting the patch possibly involves enlarging the original image by mirroring pixels. The results are summarized in row PRE-1512 Table 3. The transfer method significantly improves over the NO PRETRAIN baseline where the CNN is trained solely on the action images from Pascal VOC, without pretraining on ImageNet. In particular, we obtain best results on challenging categories `playing instrument` and `taking photo`.

In order to better adapt the CNN to the subtleties of the action recognition task, and inspired by [6], our last results were obtained by training the target task CNN without freezing the FC6 weights. More precisely, we copy the ImageNet-trained weights of layers C1...C5, FC6 and FC7, we append the adaptation layers FCa and FCb, and we retrain layers FC6, FC7, FCa, and FCb on the action recognition data. This strategy increases the performance on all action categories (row PRE-1512U in Table 3), yielding, to the best of our knowledge, the best average result published on the Pascal VOC 2012 action recognition task.

To demonstrate that we can also localize the action in the image, we train the network in a sliding window manner, as described in Section 3. In particular, we use the ground truth person bounding boxes during training, but do not use the ground truth person bounding boxes at test time. Example output maps shown in Figure 5 clearly demonstrate that the network provides an estimate of the action location in the image.

Failure modes. Top-ranked false positives in Figure 5 correspond to samples closely resembling target object classes. Resolving some of these errors may require high-level scene interpretation. Our method may also fail to recognize spatially co-occurring objects (e.g., person on a chair) since patches with multiple objects are currently excluded from training. This issue could be addressed by changing the training objective to allow multiple labels per sample. Recognition of very small or very large objects could also fail due to the sparse sampling of patches in our current implementation. As mentioned in Section 3.3 this issue could be resolved using a more efficient CNN-based implementation of sliding windows.

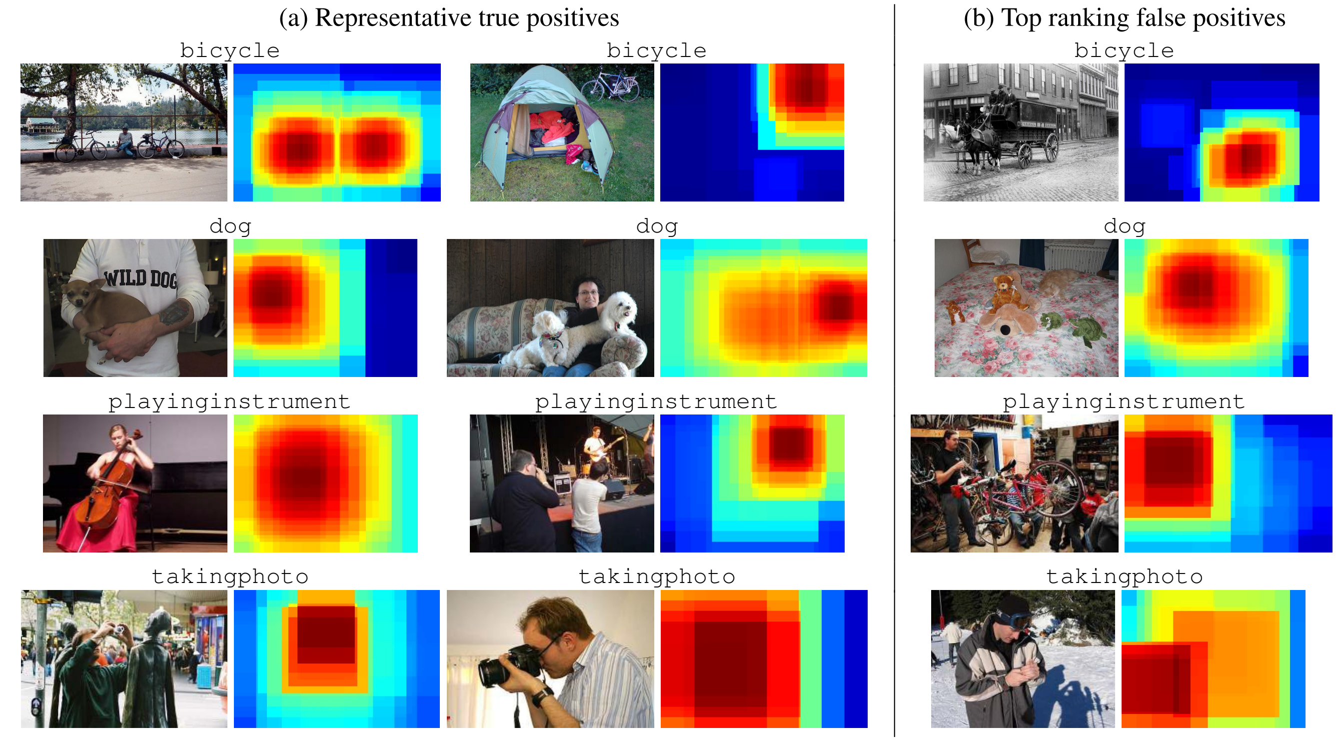

Figure 5: Response maps on representative images of several categories of the VOC 2012 object and action classification test set. The rightmost column contains the highest-scoring false positive (according to our judgement) for each of these categories. Note that correct estimates of object and action locations and scales are provided by the score maps. Please see additional results on the project webpage [2].

5. Conclusion

Building on the performance leap achieved by [25] on ILSVRC-2012, we have shown how a simple transfer learning procedure yields state-of-the-art results on challenging benchmark datasets of much smaller size. We have also demonstrated the high potential of the mid-level features extracted from an ImageNet-trained CNNs. Although the performance of this setup increases when we augment the source task data, using only 12% of the ImageNet corpus already leads to the best published results on the Pascal VOC 2012 classification and action recognition tasks. Our work is part of the recent evidence [10, 17, 43, 53] that convolutional neural networks provide means to learn rich mid-level image features transferrable to a variety of visual recognition tasks. The code of our method is available at [2].

Acknowledgements. The authors would like to thank Alex Krizhevsky for making his convolutional neural network code available. This work is partly supported by the Quaero Programme, funded by OSEO, the MSR-INRIA laboratory, ERC grant Activia, and the EIT ICT Labs.

References

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/, 2012. 6

[2] http://www.di.ens.fr/willow/research/cnn/, 2013. 6, 7

[3] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. Xing. Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks. In *ECCV*, 2008. 2

[4] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011. 2

[5] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010. 2

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537, 2011. 2, 6

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop*, 2004. 1, 2

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1, 2

[10] 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. *arXiv preprint arXiv:1310.1531*, 2013. 2, 7

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, Jun 2010. 1

[12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE PAMI*, 2013. 2, 5

[13] A. Farhadi, M.K. Tabrizi, I. Endres, and D. Forsyth. A latent model of discriminative aspect. In *ICCV*, 2009. 2

[14] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ra-

manan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 32(9):1627–1645, 2010. 1

[15] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 4

[16] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. 1

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2, 7

[18] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, CalTech, 2007. 1

[19] G.E. Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10):428–434, 2007. 2

[20] D.H. Hubel and T.N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *Journal of Physiology*, 148:574–591, 1959. 1

[21] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE PAMI*, 34(9):1704–1716, 2012.

[22] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *ACL*, 2007. 2

[23] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013. 2

[24] A. Khosla, T. Zhou, T. Malisiewicz, A.A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 2

[25] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2, 3, 4, 5, 7

[26] K.J. Lang and G.E. Hinton. A time delay neural network architecture for speech recognition. Technical Report CMU-CS-88-152, CMU, 1988. 1

[27] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 2

[28] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012. 2, 3

[29] Q.V. Le, W. Zou, S.Y. Yeung, and A.Y. Ng. Learning hierarchical spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011. 2

[30] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, Winter 1989. 1

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *PIEEE*, 86(11):2278–2324, 1998. 1

[32] Y. LeCun, L. Bottou, and J. HuangFu. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, 2004. 1

[33] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2

[34] M. Marszalek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning object representations for visual object class recognition. In *Visual Recognition Challange workshop, ICCV*, 2007. 5, 6

[35] R. Osadchy, M. Miller, and Y. LeCun. Synergistic face detection and pose estimation with energy-based model. In *NIPS*, 2005. 1

[36] S.J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010. 2

[37] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 1, 2

[38] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 2

[39] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *CVPR*, 2013. 2

[40] F. Rosenblatt. The perceptron: A perceiving and recognizing automaton. Technical Report 85-460-1, Project PARA, Cornell Aeronautical Lab, 1957. 1

[41] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. 1

[42] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 2

[43] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013. 2, 7

[44] P. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003. 1

[45] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 2

[46] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1, 2

[47] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. In *CVPR*, 2011. 5, 6

[48] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, 2010. 2

[49] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *CVPR*, 2010. 2

[50] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 2, 3, 4

[51] R. Vaillant, C. Monrocq, and Y. LeCun. Original approach for the localisation of objects in images. *IEE Proc on Vision, Image, and Signal Processing*, 141(4):245–250, 1994. 1

[52] S. Yan, J. Dong, Q. Chen, Z. Song, Y. Pan, W. Xia, Z. Huang, Y. Hua, and S. Shen. Generalized hierarchical matching for sub-category aware object classification. In *Visual Recognition Challange workshop, ECCV*, 2012. 5, 6

[53] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *Arxiv Tech Report*, abs/1311.2901, 2013. 2, 7

[54] M. Zeiler, G. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 11. 3