Deformable Part Models with CNN Features

Pierre-André Savalle[1], Stavros Tsogkas[1,2], George Papandreou[3], Iasonas Kokkinos[1,2]

[1] Ecole Centrale Paris,[2] INRIA, [3]TTI-Chicago *

Abstract. In this work we report on progress in integrating deep convolutional features with Deformable Part Models (DPMs). We substitute the Histogram-of-Gradient features of DPMs with Convolutional Neural Network (CNN) features, obtained from the top-most, fifth, convolutional layer of Krizhevsky's network [8]. We demonstrate that we thereby obtain a substantial boost in performance (+14.5 mAP) when compared to the baseline HOG-based models. This only partially bridges the gap between DPMs and the currently top-performing R-CNN method of [4], suggesting that more radical changes to DPMs may be needed.

1 Introduction

The ground-breaking results of deep learning in image classification [8] have been recently followed with equally dramatic performance improvements for object detection in the Regions with Convolutional Neural Network (R-CNN) work of [4], raising by more than 30% the detection accuracy when compared to the previous state-of-the-art.

In this work we explore to what extent these successes carry over to the Deformable Part Model paradigm, following in particular the framework layed out in [3]. By only changing the image representation, we show that the learned convolutional features yield a substantial improvement in detection performance with respect to a baseline using Histogram-of-Gradient (HOG) features [1]. This suggests that CNN features may also be beneficial in other structured prediction tasks involving DPMs, such as pose estimation [11] or facial landmark localization [12]. Our current approach however only partially bridges the performance gap between DPMs and R-CNNs; we discuss our observations and future research directions that we believe could reverse this situation.

In Section 2 we describe how we integrate CNN features into DPMs, while catering for efficiency during the learning and testing stages. In Section 3 we present results and conclude with a short discussion.

2 Integrating Convolutional Features into DPMs

Motivation and previous work: The deep convolutional network of Krizhevsky et al. [8] has been the starting point for many recent works on using CNNs for

* Workwas supported by ANR-10-JCJC-0205 (HICORE) and EU-FP7 RECONFIG.

image classification or object detection. Its first five layers are convolutional, consisting of alternating convolution and pooling layers, while the last two layers are fully connected, as in standard multi-layer perceptrons. Since the network is designed for image classification (rather than detection), its ability to localize objects was left unexplored in [8].

A first approach to object localization and detection with this class of models in the *OverFeat* system is reported by [9]. They employ a deep network trained for image classification but apply the last two fully-connected layers in a convolutional fashion to produce spatial activation responses for test images larger than the input images used for training. They jointly train two CNNs: one predicts the class of the object and another the coordinates of the bounding box containing it. At test time, they feed the network with all possible windows and scales of the original images.

Substantial improvements have been obtained by the *Regions with CNN features* (R-CNN) method [4]. They use region proposals [10] to reduce object detection to image classification. Salient regions are efficiently generated and warped to a fixed size window, which is then used as input to a CNN. Combining this idea with a network finetuning stage during training, and a bounding box regression step for better localization yields a state-of-the-art mean average precision (mAP) of 58.5% on VOC2007, and of 31.4% on ILSVRC2013. A substantial acceleration and a (moderate) further improvement in performance has been achieved in [6] by combining R-CNNs with spatial pyramid pooling.

Combining CNN features with DPMs: The region proposal strategy of R-CNN only partially captures the complexity of visual objects; in particular, for tasks such as pose estimation [11] or facial landmark localization [12], one may still need DPMs to optimize over the relationships between object parts. Using a sliding window (as in DPMs) can also potentially achieve a better recall than generic region proposal mechanisms, which may miss certain objects altogether.

A first work in this direction was presented in DenseNet [7], which proposed to compute a feature pyramid based on the topmost convolutional layers of Krizhevsky's network. In order to efficiently obtain a multi-scale representation the patchwork of scales approach [2] is used. Although [7] demonstrates how to efficiently compute feature pyramids based on a convolutional network, no quantitative evaluation on a detection task is provided.

In this work we push this line of work a step further, integrating CNN features into DPM training and testing. In particular, the standard input of Krizhevsky's network consists of a fixed-size, $224 \times 224 \times 3$ patch, which is transformed to a $13 \times 13 \times 256$ patch at the topmost (fifth) convolutional layer. Rather than working with fixed-size patches, we provide instead as input to a convolutional network an arbitrarily-sized image; following [7] we do this for multiple rescaled versions of the original image, obtaining a multi-scale CNN feature pyramid that substitutes the HOG feature pyramid typically used in DPMs. The convolutional network we use has the same architecture as the first five (convolutional) layers of Krizhevsky's but uses fine-tuned parameters from [4].

A major technical challenge is that of making the integration of CNN features with DPMs computationally efficient. Compared to using HOG features, using CNN features corresponds to an eight fold increase in the dimension (from 32 to 256), while the DPM framework is already quite computationally expensive.

To achieve efficiency during training we exploit the LDA-based acceleration to DPM training of [5], using a whitened feature space constructed for CNN features; this reduces the computation time typically by a factor of four. To achieve efficiency during convolutions with the part templates (used both during training and testing), we perform convolutions using the Fast Fourier Transform, along the lines of [2]. This reduces the convolution cost from typically 12 seconds per object (using an optimized SSE implementation) to less than 2 seconds.

A factor that turned out to be central to improving detection performance was the subsampling factor, `sub`, between the original input and the layer-5 feature representation. For Krizhevsky's network, `sub` $= 16$, meaning that a a block of 16×16 pixels in the input image is represented by a single layer-5 feature vector. As this corresponds to substantially larger bins than the ones typically used in HOG, we instead oversample our image by a factor of two before computing features, which effectively leads to `sub` $= 8$. We only report results with `sub` $= 8$, as `sub` $= 16$ leads to significantly worse APs, while `sub` $= 4$ turned out to be computationally prohibitive. We are currently exploring more efficient variants that incorporate a lower subsampling factor directly during CNN training rather than trying to make amends post-hoc.

3 Results

Our results are reported in Table 1. We consider two variants of our method: the first one, C-DPM, combines sliding window detection followed by nonmaximum suppression; the second one, C-DPM-BB, is augmented with bounding box regression, using the original bounding box coordinates as input features.

We compare these two variants to the following methods: DPMv5 refers to the baseline DPM implementation using HOG features and bounding-box regression, as in [3], while RCNN5, RCNN7, RCNN7-BB correspond to the performance of (fine-tuned) RCNN using layer 5 features, layer 7 features, or layer 7 features with an extra bounding box regression based on (richer) CNN features, respectively.

The last rows of the second and third blocks indicate the difference between the AP achieved by our method and DPMv5 or RCNN5, respectively. To have comensurate performance measures we compare DPMv5 with our variant that includes bounding box regression, (C-DPM-BB), and RCNN5, which does not include bounding box regression, to C-DPM.

From the second block it becomes clear that we significantly improve over HOG-based DPMs, while employing the exact same training pipeline; this is indicating the clear boost we obtain simply by changing the low-level image features.

However the results are not as clear-cut when it comes to comparing with RCNN. Even when comparing only to RCNN-5, we have a moderate drop in

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dtbl	dog	hors	mbike	person	plant	sheep	sofa	train	tv	mAP
C-DPM	39.7	59.5	35.8	24.8	35.5	53.7	48.6	46.0	29.2	36.8	45.5	42.0	57.7	56.0	37.4	30.1	31.1	50.4	56.1	51.6	43.4
C-DPM-BB	50.9	64.4	43.4	29.8	40.3	56.9	58.6	46.3	33.3	40.5	47.3	43.4	65.2	60.5	42.2	31.4	35.2	54.5	61.6	58.6	48.2
DPMv5	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
C-DPM-BB vs. DPMv5	+17.7	+4.1	+33.2	+13.7	+13.0	+2.6	+0.4	+23.3	+13.3	+16.4	+20.6	+30.7	+7.1	+12.3	-1.0	+19.4	+14.1	+18.4	+15.6	+15.1	+14.5
RCNN7-BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
RCNN7	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
RCNN5	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
C-DPM vs. RCNN5	-18.5	-3.8	-2.1	-2.8	+9.4	-0.4	-18.3	-5.4	+2.5	-18.7	+2.1	-1.1	0.0	-3.0	-8.4	+2.0	-19.7	+9.8	+3.0	-4.8	-3.9

Table 1. Results on PASCAL VOC 2007: average precision in percent

performance, while our DPMs are still quite behind RCNN-7. The difference with respect to RCNN-7 can be attributed to the better discriminative power of deeper features and could be addressed by incorporating nonlinear classifiers, or computing all features up to layer 7 in a convolutional manner.

But what we find most intriguing is the difference in performance between RCNN-5 and C-DPM, since both use the same features. One would expect DPMs to have better performance (since they do not rely on region proposals, and also come with many mixtures and deformable parts), but this is not the case. We suspect that this is because (i) DPMs split the training set into roughly 3 subsets (for the different aspect ratios/mixtures), effectively reducing by 3 the amount of training data and (ii) DPMs are somewhat rigid when it comes to the kind of aspect ratio that they can deal with, (3 fixed ratios) which may be problematic in the presence of large aspect ratio variatios; by contrast RCNN warps all region proposals images onto a single canonical scale. We are now working on ways to mitigate this issue.

To conclude, we have shown that replacing HOG features with CNN features yields a substantial improvement in DPM detection performance; given the widespread use of DPMs in a broad range of structured prediction tasks, e.g., [11, 12], we anticipate that this will soon become common practice.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
2. Dubout, C., Fleuret, F.: Exact acceleration of linear object detectors. In: Proc. ECCV, pp. 301–311. Springer (2012)
3. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. PAMI 32(9), 1627–1645 (2010)
4. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proc. CVPR (2014)
5. Girshick, R., Malik, J.: Training deformable part models with decorrelated features. In: Proc. ICCV. pp. 3016–3023. IEEE (2013)
6. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv preprint arXiv:1406.4729 (2014)

7. Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., Keutzer, K.: Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869 (2014)
8. Krizhevsky, A., Sutskever, I., G., Hinton: ImageNet Classification with Deep Convolutional Neural Networks. In: Proc. NIPS (2012)
9. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: International Conference on Learning Representations (ICLR 2014) (2014)
10. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV 104(2), 154–171 (2013)
11. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE Trans. Pattern Anal. Mach. Intell. 35(12) (2013)
12. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: CVPR (2012)