Crafting a multi-task CNN for viewpoint estimation

Francisco Massa, Renaud Marlet, Mathieu Aubry

Crafting a multi-task CNN for viewpoint estimation

Francisco Massa
http://imagine.enpc.fr/~suzano-f/
Renaud Marlet
http://imagine.enpc.fr/~marletr/
Mathieu Aubry
http://imagine.enpc.fr/~aubrym/

LIGM, UMR 8049,
Ecole des Ponts, UPE,
Champs-sur-Marne, France

Abstract

Convolutional Neural Networks (CNNs) were recently shown to provide state-of-the-art results for object category viewpoint estimation. However different ways of formulating this problem have been proposed and the competing approaches have been explored with very different design choices. This paper presents a comparison of these approaches in a unified setting as well as a detailed analysis of the key factors that impact performance. Followingly, we present a new joint training method with the detection task and demonstrate its benefit. We also highlight the superiority of classification approaches over regression approaches, quantify the benefits of deeper architectures and extended training data, and demonstrate that synthetic data is beneficial even when using ImageNet training data. By combining all these elements, we demonstrate an improvement of approximately 5% mAVP over previous state-of-the-art results on the Pascal3D+ dataset [28]. In particular for their most challenging 24 view classification task we improve the results from 31.1% to 36.1% mAVP.

1 Introduction

Joint object detection and viewpoint estimation is a long-standing problem in computer vision. While it was initially tackled for single objects with known 3D models [13, 18, 24], it was progressively investigated for complete object categories. The interest in this problem has recently increased both by the availability of the Pascal3D+ dataset [28], which provides a standard way to compare algorithms on diverse classes, and by the improved performance of object detection, which encouraged researchers to focus on extracting more complex information from the images than the position of objects.

Convolutional Neural Networks were recently applied successfully to this task of object category pose estimation [26, 27], leading to large improvements of state-of-the-art results on the Pascal3D+ benchmark. However many elements play an important role in the quality of these results, which have not yet been fully analyzed. In particular, several approaches have been proposed, such as a regression approach with joint training for detection [20, 21], a direct viewpoint classification [27], and a geometric structure aware fine-grained viewpoint classification [26], where the authors modify the classification objective to take into account

the uncertainty of the annotations and encode implicitly the topology of the pose space. These papers however differ in a number of other ways, such as the training data or the network architecture they use, making it difficult to compare performances. We explore systematically the essential design choices for a CNN-based approach to pose estimation and we demonstrate that a number of elements influence the performance of the final algorithm in an important way.

Contributions

In this paper, we study several factors that affect performance for the task of joint object detection and pose estimation with CNNs. Using the best design options, we rationally define an effective method to integrate detection and viewpoint estimation, quantify its benefits, as well as the boost given by deeper networks and more training data, including data from ImageNet and synthetic data. We demonstrate that the combination of all these elements leads to an important improvement over state-of-the-art results on Pascal3D+, from 31.1% to 36.1% AVP in the case of the most challenging 24 viewpoints classification. While several of the elements that we employ have been used in previous work [21, 26, 27], we know of no systematic study of their respective and combined effect, resulting in an absence of clear good practices for viewpoint estimation and sub-optimal performances. Our code is available at http://imagine.enpc.fr/~suzano-f/bmvc2016-pose/.

Related work

Convolutional Neural Networks. While convolutional neural network have a long history in computer vision (e.g. [15]), their use has been generalized only in 2012 after the demonstration of their benefits by Krizhevsky *et al*. [14] on the ImageNet large-scale visual recognition challenge [4]. Since then, they have been used to increase performances on many vision tasks.

This has been true in particular for object detection, where the R-CNN technique of Girshick *et al*. [8] provided an important improvement over previous methods on the Pascal VOC dataset [5]. Relying on an independent method to provide bounding box proposals for the objects in the image, R-CNN fine-tunes a network pre-trained on ImageNet to classify these proposal as objects or background. This method has then been improved in several ways, in particular using better network architectures [11], better bounding box proposals [23] and a better sharing of the computations inside an image [7, 10].

Viewpoint estimation. Rigid object viewpoint estimation was first tackled in the case of object instances with known 3D models, together with their detection [1, 13, 16, 17, 18, 24]. These approaches were extended to object categories detection using either extensions of Deformable Part Models (DPM) [6, 9, 12, 22], parametric models [29, 30] or large 3D instances collections [2, 26].

With the advent of Pascal3D+ dataset [28], which extends Pascal VOC dataset [5] by aligning a set of 3D CAD models for 12 rigid object classes, learning-based approaches using only on example images became possible and proved their superior performance. For example, Xiang *et al*. [28] extended the method of [22], which uses an adaptation of DPM with 3D constraints to estimate the pose. CNN-based approaches, which were until the availability of the Pascal3D+ data limited to special cases such as faces [20] and small datasets

[21], also began to be applied to this problem at a larger scale. In [19], we explored different pose representations and showed the interest of joint training using AlexNet [14] and Pascal VOC [5] data. [27] used a simple classification approach with the VGG16 network [25] and annotations for ImageNet objects and established the current state-of-the-art on Pascal3D+. [26] introduced a discrete but fine-grained formulation of the pose estimation which takes into account the geometry of the pose space, and demonstrate using AlexNet that adding rendered CAD models could improve the results over using Pascal VOC data alone.

2 Overview

We focus on the problem of detecting and estimating the pose of objects in images, as defined by the Pascal3D+ challenge Average Viewpoint Precision (AVP) metric. In particular, we focus on the estimation of the azimuthal angle. For object detection, we use the standard Fast R-CNN framework [7], which relies on region proposal but is significantly faster than the original R-CNN [8]. In addition, we associate a viewpoint to each bounding box and for each object class. Indeed, since viewpoint conventions may not be coherent for the different classes, we learn a different estimator for each class. However, to avoid having to learn one network per class, we share all but the last layer of the network between the different classes.

In Section 3, we first discuss different approaches to viewpoint prediction with CNNs and in particular the differences between regression and classification approaches. Then in Section 4, we introduce different ways to integrate the viewpoint estimation and the detection problem. Finally, in Section 5 we present the results of the different methods as well as a detailed analysis of different factors that impact performance.

Notations. We call N_s be the number of training samples and N_c the number of object classes. For $i \in \{1,...,N_s\}$ we associate to the i-th training sample x^i its azimuthal angle $\theta^i \in [0, 2\pi[$, its class $c^i \in \{1,...,N_c\}$ and the output of the network with parameters $\mathbf{w}$, $f^{\mathbf{w}}(x^i)$. The viewpoints are often discretized and we call N_v the number of bins, and $\tilde{\theta}^i \in \{1,...,N_v\}$ the bin that includes θ^i. We use subscripts to denote the elements of a tensor; for example, $f^{\mathbf{w}}(x^i)_{k,l}$ is the element (k,l) from tensor $f^{\mathbf{w}}(x^i)$.

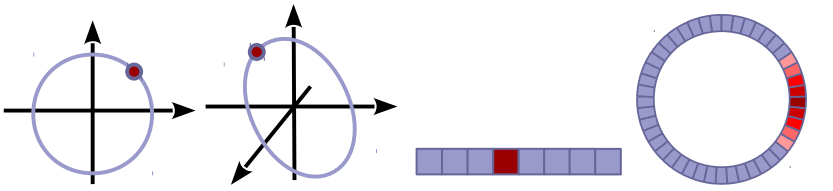
3 Approaches for viewpoint estimation

In this section, we assume the bounding box and the class of the objects are known and we focus on the different approaches to estimate their pose. Section 3.1 first discusses the design of regression approaches. Section 3.2 then presents two variants of classification approaches. The intuition behind these different approaches are visualized on Figure 1.

3.1 Viewpoint estimation as regression

The azimuth angle of a viewpoint being a continuous quantity, it is natural to tackle pose estimation as a regression problem. The choice of the pose representation $F(\theta)$ of an azimuthal angle θ is of course crucial for the effectiveness of this regression. Indeed, if we simply consider $F(\theta) = \theta$, the periodicity of the pose is not taken into account. Thus, as highlighted in [20], a good pose representation $F(\theta)$ satisfies the following properties: (a) it is invariant to the periodicity of the angle θ, and (b) it is analytically invertible.

We explore two representations which satisfy both properties:



(a) 2D regression (b) 3D regression (c) Direct classification (d) Geom. structure aware classification

Figure 1: Different approaches to orientation prediction discussed in this paper. The target for each approach is visualized in red. For the regression approaches, the possible values of the targets lie on a line. For the classification approaches, the predictions correspond to probability distributions on a discrete set.

(i) $F(\theta) = \left[\cos(\theta), \sin(\theta)\right]$, probably the simplest way to represent orientations, used for example in [21];

(ii) $F(\theta) = \left[\cos\left(\theta - \frac{\pi}{3}\right), \cos(\theta), \cos\left(\theta + \frac{\pi}{3}\right)\right]$, a formulation which was presented in [20], and that has a higher dimensionality than the previous one, allowing more flexibility for the network to better capture the pose information.

These representations have different output dimensionality N_d, respectively 2 and 3, and we designate the associated regressions by *regression 2D* and *regression 3D* respectively. Since we treat the regression independently for each class, the outputs $f^{\mathbf{w}}(x)$ of the network that we train for pose estimation have values in $\mathbb{R}^{N_c \times N_d}$ and we designate by $f^{\mathbf{w}}(x)_{c,k}$ the angular element k of the output for class c.

For training the regression with these representations, we used the Huber loss (also known as Smooth L1) on each component of the pose representation $F(\theta)$. It is known to be more robust to outliers than the Euclidean loss and provides much better results in our experiments. Our regression loss can then be written:

$$L^{\text{reg}}(\mathbf{w}) = \sum_{i=1}^{N_s} \sum_{k=1}^{N_d} H(f^{\mathbf{w}}(x^i)_{c^i,k} - F(\theta^i)_k) \tag{1}$$

with H the Huber loss. Given the output $f^{\mathbf{w}}(x)_{c,\bullet}$ of the network for a sample x of class c, we can estimate its pose simply by computing the pose of the closest point on the curve described by F (cf. Figure 1). Other regression approaches and loss are discussed in [19] but lead to lower performances.

3.2 Viewpoint estimation as classification

As pointed out by [26], the main limitation of a regression approach to viewpoint estimation is that it cannot represent well the ambiguities that may exist between different viewpoints. Indeed, objects such as a table have symmetries or near symmetries that make the viewpoint estimation problem intrinsically ambiguous, and this ambiguity is not well handled by the representations discussed in the previous paragraph. One solution to this problem is to discretize the pose space and predict a probability for each orientation bin, thus formulating the problem as one of classification. Note that a similar difficulty is found in the problem of keypoint prediction, for which the similar solution of predicting a heat map for each keypoint instead of predicting directly its position has proven successful [27].

In the case of a classification approach, the output of the network belongs to $\mathbb{R}^{N_c \times N_v}$ and each value can be interpreted as a log-probability. We write $f^{\mathbf{w}}(x)_{c,v}$ the value corresponding to the orientation bin v for an input x of class c.

3.2.1 Direct classification

The approach successfully applied in [27] is to simply predict, for each class independently, the bin in which the orientation of the object falls. This classification problem can be addressed for each object class with the standard cross-entropy loss:

$$L^{\text{classif}}(\mathbf{w}) = -\sum_{i=1}^{N_s} \log \left(\frac{\exp(f^{\mathbf{w}}(x^i)_{c^i,\tilde{\theta}^i})}{\sum_{v=1}^{N_v} \exp(f^{\mathbf{w}}(x^i)_{c^i,v})} \right) \tag{2}$$

At test time, the predicted angular bin $\hat{\theta}(x,c)$ for an input x of class c is given by

$$\hat{\theta}(x,c) = \underset{v \in \{1,\dots,N_v\}}{\arg\max} \; f^{\mathbf{w}}(x)_{c,v} \tag{3}$$

3.2.2 Geometric structure aware classification

The drawback of the previous classification approach is that it learns to predict the poses without using explicitly the continuity between close viewpoints. Two neighboring bins have indeed a lot in common. This geometrical information may be especially important for fine-grained orientation prediction, where only few examples per bin are available.

A solution to this problem was proposed in [26]. The authors finely discretize the orientations in $N_v = 360$ bins and consider the angle estimation as a classification problem, but adapt the loss to include a structured relation between neighboring bins and penalize less angle errors that are smaller:

$$L^{\text{geom}}(\mathbf{w}) = -\sum_{i=1}^{N_s} \sum_{v=1}^{N_v} \exp\left(\frac{-d(v,\tilde{\theta}^i)}{\sigma} \right) \log \left(\frac{\exp(f^{\mathbf{w}}(x^i)_{c^i,v})}{\sum_{v=1}^{N_v} \exp(f^{\mathbf{w}}(x^i)_{c^i,v})} \right) \tag{4}$$

where $d(v,\tilde{\theta}^i)$ is the distance between the centers of the two bins v and $\tilde{\theta}^i$, and σ is a parameter controlling how much similarity is enforced between neighboring bins. Following [26], we use $\sigma = 3$ for $N_v = 360$. The inference is done as in Equation (3).

4 Joint detection and pose estimation

The methods presented in the previous section assume that the object detector is already trained and kept independent from the pose estimator. Since object detection and pose estimation relies on related information, we expect a benefit from training them jointly. We thus present extensions of the methods from Section 3 to perform this joint training.

4.1 Joint model with regression

Two main approaches can be considered to extend the regression approach of Section 3.1 to jointly perform detection. The first one, described in [20] is to encode respectively the presence or absence of an object by a point close or far from the regression line described by F in the space where the regression is performed. An alternative approach, discussed

in [21], is to add an output to the regression network specifically dedicated to detection. The loss used to train the network can then be decomposed into two terms: a classification loss $L^{\text{det}}(\mathbf{w})$, which is independent on the pose, and a regression loss $L^{\text{reg}}(\mathbf{w})$ which takes into account only the pose estimation. Since state-of-the-art performance for detection are obtained using a classification loss, we selected the second option in the following.

Our network thus has two outputs: $f^{\mathbf{w},\text{det}}(x) \in \mathbb{R}^{N_c+1}$ for the detection part (predicting probabilities for each of the N_c classes and the background class), and $f^{\mathbf{w},\text{pose}}(x) \in \mathbb{R}^{N_c \times N_d}$ for the pose estimation part. The multi-task loss for joint classification and regression-based pose estimation writes as follows:

$$L^{\text{j-reg}}(\mathbf{w}) = L^{\text{det}}(\mathbf{w}) + \lambda L^{\text{reg}}(\mathbf{w}) \tag{5}$$

We define L^{reg} exactly as in Equation (1), using the pose estimation output of the network $f^{\mathbf{w},\text{pose}}(x)$. The detection loss L^{det} is the standard cross-entropy loss for detection, using the detection part of the network output $f^{\mathbf{w},\text{det}}(x)$. We set the balancing parameter $\lambda = 1$ in our experiments.

Also, we share the weights of the detection and pose estimation network only up to the *pool5* layer. This is essential to obtain a good performance, as the regression and classification losses are different enough that sharing more weights leads to much worse results.

4.2 Joint model with classification

A similar approach, separating two branches of the network, can be applied for classification. However, we introduce a new simpler and parameter-free way to perform jointly detection and pose estimation in a classification setup. Indeed, one can simply add a component, associated to the background patches, to the output vector of the pose estimation setup of Section 3.2 and normalize it globally, rather than for each class independently as in Equation (2). Each value is then interpreted as a log probability of the object being of one class and in a given orientation bin, rather than the conditional probability of the object being in a given orientation bin knowing its class. To obtain the probability of the object to belong to one class, one can simply sum the probabilities corresponding to all the bins for this class.

Similar to Section 3.2, we write $f^{\mathbf{w},\text{obj}}(x)_{c,v} \in \mathbb{R}^{N_c \times N_v}$ the value of the network output corresponding to the orientation bin v for an input x of class c. We additionally write $f^{\mathbf{w},\text{bg}}(x) \in \mathbb{R}$ its value corresponding to the background and associate a class $c^i = 0$ to the elements x^i in the background. The loss, which derives from the cross-entropy, writes:

$$\begin{aligned} L^{\text{j-classif}}(\mathbf{w}) = &- \sum_{i=1}^{N_s} \mathbb{1}_{c^i=0} \log \left(\frac{\exp(f^{\mathbf{w},\text{bg}}(x^i))}{\exp(f^{\mathbf{w},\text{bg}}(x^i)) + \sum_{c=1}^{N_c} \sum_{v=1}^{N_v} \exp(f^{\mathbf{w},\text{obj}}(x^i)_{c,v})} \right) \\ &- \sum_{i=1}^{N_s} \mathbb{1}_{c^i \neq 0} \log \left(\frac{\exp(f^{\mathbf{w},\text{obj}}(x^i)_{c^i,\tilde{\theta}^i})}{\exp(f^{\mathbf{w},\text{bg}}(x^i)) + \sum_{c=1}^{N_c} \sum_{v=1}^{N_v} \exp(f^{\mathbf{w},\text{obj}}(x^i)_{c,v})} \right) \end{aligned} \tag{6}$$

At inference, the score associated to the detection of an object x for class c is

$$S(x,c) = \frac{\sum_{v=1}^{N_v} \exp(f^{\mathbf{w},\text{obj}}(x)_{c,v})}{\exp(f^{\mathbf{w},\text{bg}}(x)) + \sum_{c'=1}^{N_c} \sum_{v=1}^{N_v} \exp(f^{\mathbf{w},\text{obj}}(x)_{c',v})} \tag{7}$$

Table 1: Different approaches for pose estimation with AlexNet architecture, Pascal VOC 2012 data, and using a fixed detector

Method	mAP	mAVP24
Regression 2D	51.6	13.9
Regression 3D	51.6	15.7
Direct classification	51.6	**19.3**
Geometric structure aware classification	51.6	18.4

5 Experiments

We now present experiments comparing the different approaches for pose estimation which were presented in the previous sections. Our experiments are based on the Fast R-CNN object detection framework [7], with Deep Mask [23] bounding boxes proposals.

We trained and evaluated our models using the Pascal3D+ dataset [28], which contains pose annotations for the training and validation images from Pascal VOC 2012 [5] for 12 rigid classes, as well as for a subset of ImageNet [4]. We also extended the training data by adding the synthetic images from [26]. The evaluation metric we used is the *Average Viewpoint Precision* (AVP) associated to Pascal3D, which is very similar to the standard Average Precision (AP) metric used in detection tasks, but which considers as positive only the detections for which the viewpoint estimate is correct. More precisely, the viewpoints are discretized into K bins and the viewpoint estimate is considered correct if it falls in the same bin as the ground-truth annotation. We focus on the AVP24 metric, which discretizes the orientation into $K = 24$ bins and is the most fine-grained of the Pascal3D+ challenge [28]. We also consider the mean AP (mAP) and mean AVP (mAVP) over all classes.

5.1 Training details

We fine-tuned our networks, starting from a network trained for ImageNet classification, using Stochastic Gradient Descent with a momentum of 0.9 and a weight decay of 0.0005. We augment all datasets with the horizontally-flipped versions of each image, flipping the target orientations accordingly. During the training of the joint detection and pose estimation models, 25% of the mini-batches consist of positive examples. Our mini-batches are of size 128 except when using synthetic images. When using synthetic images, we randomly create montages with the rendered views from [26], each montage containing 9 objects, for a total mini-batch size of 137 (96 backgrounds and 32 positive patches from real images and 9 positive synthetic objects). This allows for an efficient training in the setup of Fast R-CNN.

We initialized the learning rate at 0.001, and divided it by 10 after convergence of the training error. The number of iterations depends of the amount of training data: when using only Pascal VOC data, we decrease the learning rate after 30K iterations and continue to train until 40K; when adding ImageNet data we decrease the learning rate after 45K iterations and continue to train until 100K; and finally, when adding synthetic data, we decrease the learning rate after 100K iterations and continue to train until 300K.

All experiments were conducted using the Torch7 framework [3] and we will release our full code upon publication.

Table 2: Jointly training for detection and pose estimation with AlexNet architecture and Pascal VOC 2012 data

	Joint detector		Independant detector	
Method	mAP	mAVP24	mAP	mAVP24
Joint Regression 2D	49.2	15.7	51.6	16.4
Joint Regression 3D	49.6	17.1	51.6	17.4
Joint classification	48.6	**21.1**	51.6	**20.5**

5.2 Results

5.2.1 Comparison of the different approaches for pose estimation

We first compare the different approaches for pose estimation from Section 3. We use a fixed object detector based on the AlexNet architecture, trained for detection on Pascal VOC 2012 training set and we report the results in Table 1. We can first observe that for regression, a pose representation with a higher dimensionality (3D) performs better than when using a smaller dimensionality (2D). We believe the redundancy in the representation helps to better handle ambiguities in the estimation. The classification approach however significantly outperforms both regressions (19.3% AVP compared to 13.9% and 15.7%). Interestingly, the simplest classification approach from Section 3.2 performs slightly better than the geometry-aware method. We think the main reason for this difference is that the simple classification optimize exactly for the objective evaluated by the AVP, and thus this result can be seen as an artefact of the evaluation. Note that the results could be different for even more fine-grained estimation where less examples per class are available. Nevertheless, since the more complex geometric structure aware approach performed worse than the direct classification baseline, we focus in the rest of this paper on the simplest direct classification approach.

5.2.2 Benefits of joint training for detection and pose estimation

We evaluate the benefits of jointly training a model to detect the objects and predict their orientation. These benefits can be of two kinds. First, the order of the detections candidates given by the new detector may favor the confident orientations and thus increase the AVP. Second, the pose estimates can be better for a given object. To evaluate both effects independently, we report in Table 2 the results using both the order given by the detector used in the previous section and the order given by the new joint classifier. All experiments were performed as above, with the AlexNet architecture and the Pascal VOC training data.

Comparing Table 2 to Table 1 shows two main effects. First, the mAVP is improved even when using the same classifier, demonstrating improved viewpoint estimation with joint training. Second, the mAP is decreased, showing that the detection performs worse when trained jointly. However, one can also notice that the best mAVP is still obtained with the joint classifier. This shows that the pose estimation is better in the joint model, and also that for the case of classification the order learned when training jointly the detector favours confident poses. This is not the case for the regression approaches for which the best results are obtained using the independent detector and the jointly-learned pose estimation.

5.2.3 Influence of network architectures and training data

In this section, we consider our joint classification approach, which performs best in the evaluations of the previous section, and study how its performance varies when using different architectures and more training data.

Table 3: Influence of the amount of training data and network architecture on our joint classification approach

	AlexNet		VGG16	
Training data	mAP	mAVP24	mAP	mAVP24
Pascal VOC2012 train	48.6	21.1	56.9	27.3
+ 250 per class	51.6	25.0	58.0	30.0
+ 500 per class	53.8	26.5	59.0	31.6
+ 1000 per class	53.6	28.3	60.0	32.9
+ full ImageNet	52.8	28.4	59.9	34.4
+ synthetic data	55.9	**31.5**	61.6	**36.1**

The comparison of the left and right columns of Table 3 shows that unsurprisingly the use of the VGG16 network instead of AlexNet consistently improves performances. This improvement is slightly less for the mAVP than for the mAP, hinting that the mAVP boost is mainly due to improved detection performances.

For the training data, we first progressively add training images from ImageNet to the training images from Pascal VOC. The full subset of the ImageNet dataset annotated in Pascal3D+ contains in average approximately 1900 more images per class, but is strongly unbalanced between the different classes. The analysis of these results shows consistent improvements when the training set includes more data. Interestingly, the mAVP is improved more than the mAP, showing that the additional data is more useful for pose estimation than for detection. The addition of synthetic data (2.4M positive examples) improves the results even more, demonstrating that the amount of training data is still a limiting factor even if one uses an AlexNet architecture and includes the ImageNet images, a fact that was not demonstrated in [26]. Note that our joint approach significantly outperforms the state-of-the-art results [27] (currently 31.1% mAVP, based on VGG16 and ImageNet annotations) both without using synthetic data with VGG16, and with synthetic data and AlexNet architecture.

5.2.4 Comparison to the state of the art

Table 4 provides the details of the AVP24 performance improvements over all classes as well as a comparison with three baselines: DPM-VOC+VP [22], which uses a modified version of DPM to also predict poses, Render for CNN [26] which uses real images from Pascal VOC as well as CAD renders for training a CNN based on AlexNet, and [27] which uses a VGG16 architecture and ImageNet data to classify orientations for each object category. It can be seen that we improve consistently on all baselines except for the chair class. A more detailed analysis shows that this exception is related to the difference between the ImageNet and Pascal chairs. Indeed, when adding the ImageNet data to the Pascal data, the detection performance for chairs drops from 34.5% AP to 19.23% AP. Similarly, the difference between the very different appearance of the rendered 3D models and real images is responsible for the fact that synthetic training data decreases performance on boats, motorbikes and trains. In average, we still found that synthetic images boost the results by 1.7% mAVP.

Finaly, Table 5 provides the comparison between our full pipeline and the baselines for the 4, 8 and 16 viewpoint classification tasks, showing that our improvement of the state of the art is consistently high.

Table 4: Summary of results and comparison with baselines using AVP24

Method	aero	bike	boat	bus	car	chair	table	mbike	sofa	train	tv	mAVP24
DPM-VOC+VP [24]	9.7	16.7	2.2	42.1	24.6	4.2	2.1	10.5	4.1	20.7	12.9	13.6
Render For CNN [26]	21.5	22.0	4.1	38.6	25.5	7.4	11.0	24.4	15.0	28.0	19.8	19.8
Viewpoints & Keypoints [24]	37.0	33.4	10.0	54.1	40.0	**17.5**	19.9	34.3	28.9	43.9	22.7	31.1
Classif. approach & AlexNet	21.6	15.4	5.6	41.2	26.4	7.3	9.3	15.3	13.5	32.9	24.3	19.3
+ our joint training	24.4	16.2	4.7	49.2	25.1	7.7	10.3	17.7	14.8	36.6	25.6	21.1
+ VGG16 instead of AlexNet	26.3	29.0	8.2	56.4	36.3	13.9	14.9	27.7	20.2	41.5	26.2	27.3
+ ImageNet data	42.4	37.0	**18.0**	59.6	43.3	7.6	25.1	**39.3**	29.4	**48.1**	28.4	34.4
+ synthetic data	**43.2**	**39.4**	16.8	**61.0**	**44.2**	13.5	**29.4**	37.5	**33.5**	46.6	**32.5**	**36.1**

Table 5: Comparison with state of the art using AVP4, AVP8 and AVP16

Method	measure	aero	bike	boat	bus	car	chair	table	mbike	sofa	train	tv	mAVP
[24]	AVP4	63.1	59.4	23.0	69.8	55.2	**25.1**	24.3	61.1	43.8	59.4	55.4	49.1
Ours	AVP4	**70.3**	**67.0**	**36.7**	**75.4**	**58.3**	21.4	**34.5**	**71.5**	**46.0**	**64.3**	**63.4**	**55.4**
[24]	AVP8	57.5	54.8	18.9	59.4	51.5	**24.7**	20.5	59.5	43.7	53.3	45.6	44.5
Ours	AVP8	**66.0**	**62.5**	**31.2**	**68.7**	**55.7**	19.2	**31.9**	**64.0**	**44.7**	**61.8**	**58.0**	**51.3**
[24]	AVP16	46.6	42.0	12.7	64.6	42.7	**20.8**	18.5	38.8	33.5	42.5	32.9	36.0
Ours	AVP16	**51.4**	**43.0**	**23.6**	**68.9**	**46.3**	15.2	**29.3**	**49.4**	**35.6**	**47.0**	**37.3**	**40.6**

6 Conclusion

Combining our joint classification approach to the improvements provided by a deep architecture and additional training data, we increase state-of-the-art performance of pose estimation by 5% mAVP. We think that highlighting the different factors of this improvement and setting a new baseline will help and stimulate further work on viewpoint estimation.

Acknowlegments. This work was carried out in IMAGINE, a joint research project between Ecole des Ponts ParisTech (ENPC) and the Scientific and Technical Centre for Building (CSTB). This work was partly supported by ANR project Semapolis ANR-13-CORD-0003.

References

[1] R. Arandjelović and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *International Conference on Computer Vision (ICCV)*, 2011.

[2] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[3] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A Matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. torch.ch.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9), 2010.

[7] R. Girshick. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[9] Daniel Glasner, Meirav Galun, Sharon Alpert, Ronen Basri, and Gregory Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *International Conference on Computer Vision (ICCV)*, 2011.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision (ECCV)*, pages 346–361, 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[12] M. Hejrati and D. Ramanan. Analyzing 3D objects in cluttered images. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[13] D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *International Conference on Computer Vision (ICCV)*, 1987.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] 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, 1989.

[16] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3D point clouds. In *European Conference on Computer Vision (ECCV)*, 2012.

[17] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing Ikea objects: Fine pose estimation. In *International Conference on Computer Vision (ICCV)*, pages 2992–2999. IEEE, 2013.

[18] D. Lowe. The viewpoint consistency constraint. *International Journal of Computer Vision (IJCV)*, 1(1):57–72, 1987.

[19] Francisco Massa, Mathieu Aubry, and Renaud Marlet. Convolutional neural networks for joint object detection and pose estimation: A comparative study. *arXiv preprint arXiv:1412.7190*, 2014.

[20] M. Osadchy, Y. LeCun, and M. L. Miller. Synergistic face detection and pose estimation with energy-based models. *The Journal of Machine Learning Research (JMLR)*, 8: 1197–1215, 2007.

[21] H. Penedones, R. Collobert, F. Fleuret, and D. Grangier. Improving object classification using pose information. Technical report, Idiap Research Institute, 2011.

[22] Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Teaching 3D geometry to deformable part models. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3362–3369, 2012.

[23] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 1990–1998. Curran Associates, Inc., 2015.

[24] L. Roberts. Machine perception of 3-D solids. In *PhD. Thesis*, 1965.

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

[26] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. In *International Conference on Computer Vision (ICCV)*, pages 2686–2694, 2015.

[27] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1510–1519, 2015.

[28] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond Pascal: A benchmark for 3D object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[29] Yu Xiang and Silvio Savarese. Estimating the aspect layout of object categories. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[30] M. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3D representations for object recognition and modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.