Is 2D Information Enough For Viewpoint Estimation?

Amir Ghodrati
amir.ghodrati@esat.kuleuven.be
Marco Pedersoli
marco.pedersoli@esat.kuleuven.be
Tinne Tuytelaars
tinne.tuytelaars@esat.kuleuven.be

KU Leuven, ESAT - PSI, iMinds
Leuven, Belgium

Abstract

Recent top performing methods for viewpoint estimation make use of 3D information like 3D CAD models or 3D landmarks to build a 3D representation of the class. These 3D annotations are expensive and not really available for many classes. In this paper we investigate whether and how comparable performance can be obtained without any 3D information. We consider viewpoint estimation as a 1-vs-all classification problem on the previously detected object bounding box. In this framework we compare several features and parameter configurations and show that the modern representations based on Fisher encoding and convolutional neural network based features together with a neighbor viewpoints suppression strategy on the training data lead to comparable or even better performance than 3D methods.

1 Introduction

Estimating the pose of objects is a classical problem in vision. It aims at predicting a discrete or continuous viewpoint. In conjunction with object detection, viewpoint estimation is receiving increasing attention lately. Recent trends in the vision community suggest that, for an accurate estimation of the object pose, 3D information about the object class is beneficial. For instance, Pepik *et al*. [25] show that using 3D CAD models of the class of interest can lead to a 3D representation of a deformable part model which, even though it has slightly worse detection performance, obtains state-of-the-art results in terms of pose estimation. Likewise, Hejrati and Ramanan [13] show that providing 3D landmarks of cars can lead to a very accurate estimation of their 3D pose.

However, 3D information (either 3D CAD models or 3D landmarks) is expensive to obtain and not available for many classes. In this paper, we show that a very simple 2D architecture (in the sense that it does not make any assumption or reasoning about the 3D information of the object) generally used for object classification, if properly adapted to the specific task, can provide top performance also for pose estimation.

More specifically in this work, we demonstrate on several datasets how a 1-vs-all classification framework based on a Fisher Vector (FV) pyramid and with neighbor viewpoints suppression (see sect. 3) can be used for pose estimation. Furthermore, we investigate the

performance of our system substituting the FV representation by the features extracted from a convolutional neural network (CNN) that recently has obtained very impressive results on the object classification task [6, 16, 27]. Our results show that for the fine-grained task of pose estimation both representations perform equally well and similarly or better than 3D methods previously proposed and designed specifically for the problem of pose estimation.

The paper is organized as follows. In section 2 we relate our method with the current literature in 2D/3D pose estimation. In section 3 we explain each component of our method and how those components interact with each other. Finally experiments are presented in section 4. Conclusions are drawn and future work is discussed in section 5.

2 Related Work

There are two lines of research for viewpoint estimation: one that uses 2D models for the pose representation and the other that leverages on 3D information to tackle the problem. Inspired by successes of deformable part models, several works have built 2D viewpoint-dependent detectors. Typically, they explicitly handle viewpoints and discriminatively train models where the number of components corresponds to the views of an object for joint viewpoint classification and object detection. These models vary from rigid HOG templates [22] to deformable part models [12, 19, 36]. The drawback of such formulations is that they typically require training and evaluating a large number of view-based detectors which can be computationally quite demanding. Recently Redondo-Cabrera *et al*. [28] proposed a Hough Forest based method for simultaneous object detection and continuous pose estimation. However their detection performance is not as good as DPM-based methods.

Latest progresses on pose estimation have mostly utilized 3D CAD models [18, 24, 25, 31]. Pepik *et al*. [24] introduce a 3D extension of the deformable part model where part appearances as well as spatial deformations are represented in 3D. Such formulation allows synthesizing part appearance models for arbitrary viewpoints. Similarly, Zia *et al*. [37] first obtain a rough localization and pose of the object by using an off-the-shelf method and then a continuous pose is estimated by using annotated 3D CAD models. Arie-Nachimson and Basri [1] and Glasner *et al*. [10] use pose estimation prior to construct a 3D point cloud of object instances from training images. This limits their methods to datasets where such reconstruction is possible. Hejrati and Ramanan [13] estimate car poses using an explicit 3D model of shape and viewpoint which is learned from structure-from-motion (SFM). The drawback of such methods is that they require labeled landmark positions of training data which is expensive to collect. Sun *et al*. [33] and Su *et al*. [32] build 3D pose models by adopting the strategy of grouping local features into parts and learn part locations across viewpoints using generative models. Finally, Fanelli *et al*. [7] showed the usefulness of depth information for solving the problem of head pose estimation. They learn a mapping between simple depth features to 3D nose coordinates and rotation angles and estimate head pose through random forest based classifiers. In general, methods that rely on 3D models are not easy to collect for certain object categories.

Recently great interest has been expressed in fisher kernel and convolutional neural network representations which have shown outstanding performance on several vision tasks. Simonyan *et al*. [30] showed that Fisher vectors on densely sampled SIFT features are capable of achieving state-of-the-art face verification performance. Recently, Toshev and Szegedy [34] propose a cascade of deep neural network regressors that aim to predict articulated human body joints. Jain *et al*. [15] trained multiple convolutional neural nets to

perform independent detection of parts. However, up to our knowledge, there is no work that uses Fisher vector or convolutional neural network features in a simple 2D representation for the task of viewpoint estimation.

3 Proposed Method

Our method takes as input a detection bounding box, extracts features and assigns to the bounding box a pose. The estimation of the pose is done with a one-vs-all classifier of a discrete set of viewpoints. In the rest of this section, we explain in detail each step of our pipeline.

Detection. For detection, we use the de-facto standard detector based on deformable part models (DPM) [8, 9]. For each image, we apply the detector at multiple scales and collect detections which later are processed for estimating their poses. For both training and testing, we train our model on the detected objects (i.e. we did not use ground-truth bounding-boxes for training pose models) since we want the data to be generated from the same distribution.

Feature Extraction. In all FV-based experiments we extract dense SIFT descriptors [20] from the output of the detector. Specifically, we extract features over 5 scales, with a scaling factor of $\sqrt{2}$. 32×32 pixels patches are sampled with step size of 5 pixels from every detected bounding box. We call this basic feature representation `sift`. In addition, as proposed by Carreira *et al*. [3], we also repeat the experiments with enriched SIFT descriptors where the it is enriched with the location of the patch centre with respect to the upper-left corner of the bounding box, normalized by its size. In this case, for each patch, the final descriptor is a $L1$-normalised concatenation of the SIFT descriptor and the patch location (`sift+loc`), resulting in a 130-dimensional descriptor.

Fisher Vector. FV [14] encodes information about the generative model that produces the low-level features by computing the gradient of the feature samples with respect to the model parameters. For computing the FV, we use the improved procedure proposed by Perronnin *et al*. [26] where a Gaussian Mixture Model (GMM) is fitted to dimensionality reduced SIFT features.

Specifically, after reducing the feature dimensions to 60 using PCA, we estimate $\lambda_i = \{w_i, \mu_i, \Sigma_i, i = 1..K\}$, the parameters of the GMM, on a $100K$ sampled features set where w_i, μ_i, Σ_i are weight, mean, and diagonal covariance of the i-th mixture model respectively and K is the number of mixtures. Afterwards we estimate first and second order gradient statistics of each feature by computing its derivative w.r.t. the Gaussian means and variances. Let $G^X_{\lambda_i}$ be the weighted average of low-level features statistics with respect to component i,

$$G^X_{\lambda_i} = \frac{1}{T} \sum_{t=1}^{N} \alpha_i^t \psi(x_t; \lambda_i),$$

$$\psi(x_t; \lambda_i) = [\frac{1}{\sqrt{w}}(\frac{x_t - \mu_i}{\sigma_i}), \frac{1}{\sqrt{2w}}(\frac{(x_t - \mu_i)^2}{{\sigma_i}^2} - 1)].$$

Then each image is represented by stacking $G^X_{\lambda_i}$ for all mixtures: $[G^X_{\lambda_1}, ..., G^X_{\lambda_K}]$, where $X = \{x_t, t = 1..N\}$ is the set of N low-level features extracted from the image and α_i^t is

the soft weight of the t-th feature for the i-th Gaussian. Following [26], we further normalise Fisher vectors by signed square-rooting and then $L2$ normalisation. In our experiments, we built $K = 128$ Gaussian mixtures which leads to an image representation of length $2Kd = 2 \times 128 \times 60 = 15360$.

Spatial Information. It is known that in Fisher encoding, the spatial layout of the appearance is completely ignored (except when using the `sift+loc` enriched features). Without doubt, the spatial information may convey useful cues for pose estimation. In this paper, we encode spatial information in two ways. First as a low-level strategy by augmenting SIFT with location of the patch (as previously described) and second by building a Spatial Pyramid [17] on top of the FV. In the experiments we use a spatial pyramid that divides the bounding box into 4×4, 2×2 and 1×1 cells and then stacks the FVs computed for each cell separately. We call this configuration `fisher+spm`. As we will see in section 4.2, the two spatial encodings are complementary.

Learning. We want to transform the discrete viewpoint estimation problem to classification. To do so, we consider each viewpoint as a different class. Then, for each viewpoint we learn a linear SVM based on a 1-vs-all strategy. In this scenario an important difference with a standard multiclass problem is that nearby viewpoints are generally visually very correlated. In this sense, it is not reasonable to assign to all negative poses equal importance. In the experimental results we show that eliminating nearby poses from negative samples always improve the viewpoint estimation. We call this procedure neighboring viewpoint suppression or briefly `nv-suppression`. For very coarse binning, since it might happen that too much negative data is suppressed, whenever continuous pose is provided, we suppress negative data only up to 10 degrees apart from the positive samples. Note that this is similar to the recently proposed one-vs-most technique of Berg *et al*. [2].

Convolutional Neural Networks. We also tested Deep Convolutional Activation Feature (`decaf`) [5] on our framework to evaluate how good recently popular deep-learning approaches perform on viewpoint estimation. Decaf is based on the deep convolutional neural network architecture proposed by Krizhevsky *et al*. [16], which won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 [4]. In `decaf` the neurons activation of the late hidden layers of a pre-trained network are used as strong features for generic vision tasks with impressive results [5]. We use the model pre-trained on ILSVRC since in pose estimation there are too few training samples to properly learn a full deep representation from scratch. The final network contains five convolutional layers followed by three fully-connected layers named by layer from 1 to 8. We refer to [16] for a detailed discussion of the architecture. For pose training, we first extract the $L2$-normalized features from the pooled output of layer 5 (last convolutional layer) and then use the same learning strategy as explained above for `fisher`. In this case, the pose representation has 9216 feature dimensions.

4 Experimental Evaluation

In this section we first describe the characteristics of the four datasets that we use and then on these datasets we thoroughly evaluate and compare several state-of-the-arts methods based on 2D and 3D information for estimating poses.

4.1 Datasets

Faces. We train the detector on a subset of the CMU MultiPIE face dataset [11] and test it on the Annotated face-in-the-wild [36] (AFW) dataset. The CMU MultiPIE face dataset contains around 75000 images of 337 people over 13 viewpoints spanning over 180 degrees discretized every 15 degrees, with different illumination conditions and expressions. As in [36], in our experiments, we use 900 faces for training with 300 of those frontal and the rest evenly distributed among other viewpoints. AFW test set contains 468 faces from 205 images and 13 discretized viewpoints. Images contain cluttered backgrounds with large variations in face appearance. The metric used for this dataset is the same as in [36] and reports the fraction of faces for which the estimated pose is within some error tolerance (± 15 and ± 30). Notice that, to make the evaluation more realistic, missed detections are counted as errors in pose estimation. In the following tables, we call this evaluation fraction of valid poses (FVP).

Cars. We also present results on EPFL Multi-view car dataset [22]. It contains 2299 images of 20 different car models. Cars are rotated over 360 degrees and their continuous viewpoint angle can be approximately calculated using the capture time of each image and the frontal view capturing time information that is provided. Images are captured using a static camera and all cars appear in the center of the image, without occlusions. Since a continuous pose is provided for this dataset, in our evaluation we divide the viewpoints into 8, 16 and 36 discrete bins. We follow the experimental setup of [22] and use the first 10 sequences for training and next 10 sequences for testing. The evaluation metric that we use for this dataset is Mean Precision of Pose Estimation (MPPE) [19] and Median Angular Error (MAE). MPPE is computed as the average of the diagonal of the confusion matrix and MAE is the median error, where the error is measures computed as $min\{|\theta - \theta^*|, 360 - (|\theta - \theta^*|)\}$ with θ the estimated viewpoint angle and θ^* the ground truth viewpoint.

General Objects. Finally, we evaluate our methods on two general objects datasets: PASCAL3D+ [35] and a subset of the 3DObject dataset [29]. PASCAL3D+ augments 12 rigid categories of the PASCAL VOC 2012 [6] with 3D annotations. For each category more images are added from ImageNet [4] and on average there are more than 3000 object instances per category. Since [35] reported the baseline using the `train` subset of PASCAL VOC 2012 (detection challenge) for training and `val` subset for evaluation, we follow the same protocol. For evaluation, we use the Average Viewpoint Precision (AVP). AVP takes into account the detection performance in the evaluation of the pose estimation. In this way, an output from the detector is considered to be correct if and only if the bounding box overlap with the ground truth annotation is larger than 50% *and* the viewpoint is correct. As a result, AP is an upper bound for AVP. Note that Pose Estimation Average Precision (PEAP) proposed in [19] is different from AVP. PEAP[19] uses precision and recall of pose estimation whereas, even if not formally specified in [35], AVP uses precision of pose estimation and recall of detection.

Finally, the 3DObject dataset contains 10 everyday object classes such as iron, car and stapler. Each category includes 10 instances observed from 8 different viewpoints. Because other papers that use 3D information have published their results only on car and bicycle categories, we evaluate on car and bicycle as well. We follow the testing protocols of [29] and report results in terms of MPPE for this dataset.

4.2 Experimental Evaluation

Detection. In all experiments the first part of our algorithm consists of detecting the object of interest. For this we use standard DPM (`voc-release5`) [9] with 6 components. For faces, we train DPM on 900 images of MultiPIE where the components are initialized based on the face orientation. We evaluate on AFW obtaining an AP of 88.3% with a maximum recall of 98.1%. Note that recently Mathias *et al.* [21] reported significantly better detection performance on this dataset. For detection on EPFL cars, we use the PASCAL VOC 2007 pre-trained DPM car model (`voc-release5`) [9]. Its AP is 88.2% with a maximum recall of 100% on test images. For PASCAL3D+ dataset, following [35], we train DPM (in this case we use version `voc-release4.01` to be compatible with the original results) on the `train` subset of PASCAL VOC 2012 and evaluate on `val`. The AP is reported in table 6 for each object category. For detection on Object3D cars and bicycles, we again use DPM car and bicycle models pre-trained on PASCALVOC-2007 (`voc-release5`) [9]. Their APs on test data are 88.9% and 79.2% respectively with maximum recall of 100% and 96.8%.

Pose Estimation. In table 1, we evaluate the performance of different features and encoding on the EPFL car dataset using 8 view models, each covering 45 degrees and AFW using the 13 views learned on MultiPIE. We evaluate the methods either with `ground-truth` bounding boxes or with the bounding boxes obtained from a `detector` applied to all the images (both training and test). In general, as expected the ground truth bounding boxes give better results, but there are also some exceptions. `decaf` trained with ground-truth bounding boxes is among the best on both datasets, but when using the detector bounding boxes the performance drops significantly. In contrast `fisher` is less sensitive to the bounding box localization. Considering the absolute performance of the different methods, we clearly notice that the baseline based on bag-of-words `BoW` (dictionary of size 4000 and the final representation is $L2$-normalized) is the poorest method for pose representation. The best representation on both datasets is `fisher` with spatial pyramid `spm`. Comparing `sift` and `sift+loc`, we can see that also embedding spatial information in the low-level representation is still advantageous for pose estimation. Also the performance of `decaf` is quite good, especially considering its much lower dimensionality. Based on these conclusions, we select the two last methods for the next experiments.

		EPFL		AFW	
Feature Type	Encoding	MPPE (`ground-truth`)	MPPE (`detector`)	FVP±15 (`ground-truth`)	FVP±15 (`detector`)
`sift`	`BoW`	56.6%	54.8%	43.4%	49.4%
`sift`	`fisher`	68.4%	68.2%	51.1%	54.3%
`sift`	`fisher+spm`	82.1%	80.1%	73.3%	69.7%
`sift+loc`	`fisher+spm`	**82.8%**	**81.8%**	75.8%	**70.3%**
`decaf`	-	77.2%	72.0%	**77.3%**	67.9%

Table 1: An evaluation with training and testing data from ground-truth bounding boxes (3rd and 5th columns) and output of detector (4th and 6th columns) on the EPFL car dataset and AFW faces dataset. MPPE is computed as the average of the diagonal of the confusion matrix. FVP±15 is the fraction of faces that are within ±15 error interval, counting missed detections as infinite error.

		EPFL			AFW	
Method	nv-suppression	8 bins	16 bins	36 bins	FVP±15	FVP±30
fisher+spm	×	**81.8%**	71.2%	46.4%	70.3%	84.2%
fisher+spm	✓	80.6%	**72.2%**	**51.8%**	**78.6%**	**90.6%**
decaf	×	72.0%	62.1%	39.1%	67.9%	82.3%
decaf	✓	**76.6%**	**67.8%**	**45.9%**	**86.5%**	**93.4%**

Table 2: The effect of suppressing negative neighboring viewpoints samples. On EPFL car dataset, MPPE is computed . Last two columns are fraction of faces that are within ±15 and ±30 error interval respectively, counting missed detections as infinite error for AFW dataset.

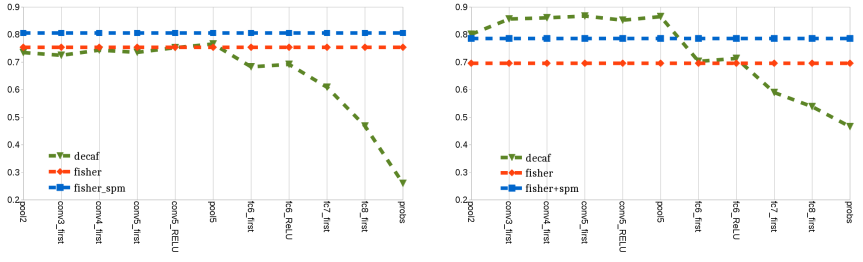

Figure 1: Evaluation of 8-bins pose estimation problem on EPFL car dataset (left) and AFW face dataset (right) for different layers of CNN network. Horizontal lines are Fisher vector performances.

Effect of neighbor samples. In table 2 we investigate the impact of `nv-suppression` of negative samples explained in sect. 3. On EPFL, for the coarsest binning (8 bins), the suppression scheme does not help, probably because the confusion between nearby poses in coarse binning is not an issue. However, when using a finer binning, the advantage of the `nv-suppression` is quite evident. Consequently, we continue our experiments with the suppression of the nearest neighbors enabled.

Decaf. Next, we investigate the impact of features obtained from different layers of a convolutional neural network for the task of pose estimation. To this end, we select the output of different layers as features and compute the performance for every layer as shown in figure 1. Note that we use 8 bins for the EPFL car dataset and apply the `nv-suppression` on negative samples on all `fisher`, `fisher+spm` and `decaf`. On the EPFL car dataset and the PASCAL3D+ dataset (table 6), `fisher+spm` outperforms `decaf` but for the AFW faces that with 13 different poses, `decaf` performs better. In addition, as it is shown, the last convolutional layer (layer 5) outperforms the others in both datasets. Finally, it is interesting to notice that the lower convolutional layers perform quite well whereas in other tasks generally they do not perform good.

Computational Cost. Another advantage of the proposed method is the reduced computational cost. Although a precise evaluation for each method in terms of time is difficult to obtain we can still reason about the computational cost of the different methods. We can safely claim that all the methods based on DPM are computationally more demanding

bins	[23]	[19]	3D2PM-C Lin [24]	3D2PM-D [24]	ours(fisher+spm)	ours(decaf)
8 bins	-	73.7%	78.3%	78.5%	**80.6%**	76.6%
16 bins	41.6%	66.0%	69.0%	69.8%	**72.2%**	67.8%
36 bins	-	-	**52.1%**	45.8%	51.8%	45.9%

Table 3: Comparison with state of the art viewpoint classification methods on the EPFL dataset.

bins	3D2PM-C Lin [24]	3D2PM-D [24]	[10]	ours(fisher+spm)	ours(decaf)
8 bins	**11.1**	12.9	24.8	12.5	13.5
16 bins	6.9	7.2	-	**6.75**	7.75
36 bins	**4.7**	5.8	-	5.0	6

Table 4: Viewpoint estimation in terms of MAE for EPFL car dataset.

than ours. This is due to the fact that in our method we use standard DPM models with 6 components and the following step, based on FV or Decaf has a negligible cost. For example, extracting SIFT, building a pyramid of Fisher vector and a 36-bins pose classification takes on average 1.38, 0.73 and 0.19 seconds respectively on a quad-core processor using MATLAB on the EPFL dataset. Extracting decaf features for an image takes on average 0.2 seconds while the training time for 36 one-vs-all SVM linear models for fisher+spm and decaf is 290 and 6 seconds respectively. Instead, other methods generally use a DPM component for each view, so that, especially when searching for fine pose estimation, the computational cost will be higher (e.g. detection using the standard DPM takes around 4 seconds for each EPFL image, while with 36 bins the computational cost of viewpoint-DPM should be 6 times the standard DPM model).

Comparison with state-of-the-art. Table 3 compares our method to other state-of-the-art methods on the EPFL car dataset. fisher+spm outperforms all methods including 3D models on this dataset for 8 and 16 viewpoint bins and is slightly worse than the continuous model of [24] but outperforms their discrete version. decaf could not obtain state of the art performance on this dataset but it is on par with the discrete model of [24].

For EPFL car dataset, as the angular viewpoint annotations are provided, we can also use the Median Angular Error for evaluation. Note that MAE is a metric for evaluation of continuous estimation but we are using a discrete estimation. Thus, for each bin, we assume the center of the bin as the estimated angular viewpoint. In terms of MAE, as shown in table 4, fisher+spm outperforms state-of-the-art discrete models (3D2PM-D) that use 3D CAD information and is on par with continuous appearance models (3D2PM-C Lin).

Table 5 shows the results of our methods and the current state-of-the art on the AFW dataset. Within ± 30 degree error tolerance, fisher+spm and decaf both perform well and outperform all the other methods whereas with ± 15 degrees error tolerance, decaf outperforms all other methods. These results are quite important especially considering that [36] is tuned for face detection and pose estimation problems while fisher+spm and decaf are applicable to any other category.

For PASCAL3D+ dataset, the results of our methods and methods of [35], [25] are shown in Table 6. Same as [35], we ignore the bottle category since its instances are often symmet-

[1] From Zhu *et al.* [36]

method	±15	±30
Face.com[1]	64.3%	86.5%
[36]- indp. model	81.0%	89.0%
[36]- shared. model	76.9%	87.0%
Multi-HoG[1]	74.6%	85.0%
ours(`fisher+spm`)	78.6%	90.6%
ours(`decaf`)	**86.5%**	**93.4%**

Table 5: Comparison with state of the art pose estimation methods on the AFW face dataset.

AP/AVP	aeroplane	bicycle	boat	bottle	bus	car	chair	diningtable	motorbike	sofa	train	tvmonitor	avg.
[35]-4V	40.0/34.6	45.2/41.7	3.0/1.5	-/-	49.3/26.1	37.2/20.2	11.1/6.8	7.2/3.1	33.0/30.4	6.8/5.1	26.4/10.7	35.9/34.7	26.8/19.5
[25]-4V	41.5/37.4	46.9/43.9	0.5/0.3	-/-	51.5/48.6	45.6/36.9	8.7/6.1	5.7/2.1	34.3/31.8	13.3/11.8	16.4/11.1	32.4/32.2	27.0/23.8
ours(`fisher+spm`)-4V	40.1/26.7	48.0/34.4	6.1/2.3	-/-	54.1/50.7	36.1/28.9	14.8/11.1	9.1/5.4	32.9/29.4	18.9/17.3	36.1/32.5	33.2/26.9	29.9/**24.1**
ours(`decaf`)-4V	40.1/24.5	48.0/32.9	6.1/2.4	-/-	54.1/49.6	36.1/24.1	14.8/10.7	9.1/6.1	32.9/27.6	18.9/14.2	36.1/32.2	33.2/27.6	29.9/22.9
[35]-8V	39.8/23.4	47.3/36.5	5.8/1.0	-/-	50.2/35.5	37.3/23.5	11.4/5.8	10.2/3.6	36.6/25.1	16.0/12.5	28.7/10.9	36.3/27.4	29.9/18.7
[25]-8V	40.5/28.6	48.1/40.3	0.5/0.2	-/-	51.9/38.0	47.6/36.6	11.3/9.4	5.3/2.6	38.3/32.0	13.5/11.0	21.3/9.8	33.1/28.6	28.3/21.5
ours(`fisher+spm`)-8V	40.1/23.6	48.0/27.6	6.1/2.4	-/-	54.1/50.3	36.1/26.6	14.8/9.1	9.1/6.0	32.9/24.7	18.9/16.9	36.1/31.3	33.2/26.5	29.9/**22.3**
ours(`decaf`)-8V	40.1/17.7	48.0/27.7	6.1/1.9	-/-	54.1/49.6	36.1/23.3	14.8/7.8	9.1/4.8	32.9/27.1	18.9/11.1	36.1/31.2	33.2/26.4	29.9/20.8
[35]-16V	43.6/15.4	46.5/18.4	6.2/0.5	-/-	54.6/46.9	36.6/18.1	12.8/6.0	7.6/2.2	38.5/16.1	16.2/10.0	31.5/22.1	35.6/16.3	30.0/15.6
[25]-16V	38.0/15.9	45.6/22.9	0.7/0.3	-/-	55.3/49.0	46.0/29.6	10.2/6.1	6.2/2.3	38.1/16.7	11.8/7.1	28.5/20.2	30.7/19.9	28.3/**17.3**
ours(`fisher+spm`)-16V	40.1/16.3	48.0/18.0	6.1/1.5	-/-	54.1/42.9	36.1/19.6	14.8/7.4	9.1/4.6	32.9/15.9	18.9/13.8	36.1/29.0	33.2/21.5	29.9/**17.3**
ours(`decaf`)-16V	40.1/11.2	48.0/19.5	6.1/1.7	-/-	54.1/43.5	36.1/19.4	14.8/6.3	9.1/4.6	32.9/20.6	18.9/10.5	36.1/28.6	33.2/22.3	29.9/17.1
[35]-24V	42.2/8.0	44.4/14.3	6.0/0.3	-/-	53.7/39.2	36.3/13.7	12.6/4.4	11.1/3.6	35.5/10.1	17.0/8.2	32.6/20.0	33.6/11.2	29.5/12.1
[25]-24V	36.0/9.7	45.9/16.7	5.3/2.2	-/-	53.9/42.1	42.1/24.6	8.0/4.2	5.4/2.1	34.8/10.5	11.0/4.1	28.2/20.7	27.3/12.9	27.1/13.6
ours(`fisher+spm`)-24V	40.1/12.3	48.0/12.6	6.1/1.3	-/-	54.1/40.2	36.1/15.9	14.8/5.5	9.1/4.6	32.9/13.2	18.9/10.2	36.1/20.4	33.2/15.0	29.9/**13.7**
ours(`decaf`)-24V	40.1/10.0	48.0/12.9	6.1/0.8	-/-	54.1/39.8	36.1/16.7	14.8/4.9	9.1/4.5	32.9/13.4	18.9/7.4	36.1/21.7	33.2/18.9	29.9/**13.7**

Table 6: The results of [35], [25] and ours for 4, 8, 16 and 24 viewpoint angles respectively on PASCAL3D+ dataset. The first number is AP of object detection and the second one is AVP of pose estimation.

category	[23]	[19]	[25]	[10]	[24]	[18]	ours(`fisher+spm`)	ours(`decaf`)
cars	86.1%	89.0%	**97.9%**	85.3%	95.8%	70.0%	95.8%	**97.9%**
bicycle	80.8%	90.0%	**98.9%**	-	96.0%	75.5%	98.1%	86.1%

Table 7: Viewpoint estimation on car and bicycle classes from Object3D dataset (MPPE).

ric across different viewpoints. We notice the same trend as in the previous experiments: `fisher+spm` performs best on all viewpoint angles. `decaf` results are slightly lower but still comparable with [25] which relies on 3D CAD models. For more classes like train or sofa, our method performs markedly better than [25], whereas for other classes, like bicycle and car, [25] performs better. We believe this is correlated with the fact that for the latter classes, more and better 3D CAD models are available and therefore a better 3D representation can be learned.

For cars and bicycles of the 3D Object dataset for which objects are provided in 8 different poses, as shown in table 7, both `decaf` and `fisher+spm` again outperform most of the other methods in the literature and achieve competitive performance to methods that use 3D CAD data ([25] and [24]).

5 Conclusion

In this paper, we have presented a study of different methods for pose estimation on four well-known and challenging datasets. Through an extensive evaluation we can clearly see

that, in contrast to common believe, the very simple framework based on the extraction of features on the object bounding box using modern features (`decaf`) or in combination with modern encodings (`fisher+spm`) can in most of the cases outperform the state-of-the-art including methods based on 3D or much more complex and computationally expensive models. This suggests that the next generation of pose estimation methods should probably combine these powerful 2D representations with 3D reasoning.

6 Acknowledgments

This work was supported in part by DBOF PhD scholarship and FP7 ERC Grant 240530 COGNIMUND.

References

[1] Mica Arie-Nachimson and Ronen Basri. Constructing implicit 3d shape models for pose estimation. In *ICCV*, 2009.

[2] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle Alexander, David Jacobs, and Peter Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, 2014.

[3] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012.

[4] Jia Deng, Alex Berg, Sanjeev Satheesh, Hao Su, Aditya Khosla, and Li Fei-Fei. Imagenet large scale visual recognition challenge 2012. URL http://www.image-net.org/challenges/LSVRC/2012/.

[5] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2), 2010.

[7] Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *CVPR*, 2011.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on PAMI*, 2010.

[9] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/.

[10] Daniel Glasner, Meirav Galun, Sharon Alpert, Ronen Basri, and Gregory Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011.

[11] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5), 2010.

[12] Chunhui Gu and Xiaofeng Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010.

[13] Mohsen Hejrati and Deva Ramanan. Analyzing 3d objects in cluttered images. In *NIPS*, 2012.

[14] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. *NIPS*, 1999.

[15] Arjun Jain, Jonathan Tompson, Mykhaylo Andriluka, Graham W Taylor, and Christoph Bregler. Learning human pose estimation features with convolutional networks. In *CVPR*, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[17] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[18] Joerg Liebelt and Cordelia Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, 2012.

[19] Roberto Javier Lopez-Sastre, Tinne Tuytelaars, and Silvio Savarese. Deformable part models revisited: A performance evaluation for object category pose estimation. In *ICCV Workshops*, 2011.

[20] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[21] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. Face detection without bells and whistles. In *ECCV*, 2014.

[22] Mustafa Ozuysal, Vincent Lepetit, and Pascal Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.

[23] Nadia Payet and Sinisa Todorovic. From contours to 3d object detection and pose estimation. In *ICCV*, 2011.

[24] Bojan Pepik, Peter Gehler, Michael Stark, and Bernt Schiele. 3d2pm–3d deformable part models. In *ECCV*, 2012.

[25] Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Teaching 3d geometry to deformable part models. In *CVPR*, 2012.

[26] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

[27] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014.

[28] C. Redondo-Cabrera, R. Lopez-Sastre, and T. Tuytelaars. All together now: Simultaneous object detection and continuous pose estimation using a hough forest with probabilistic locally enhanced voting. In *BMVC*, 2014.

[29] Silvio Savarese and Fei-Fei Li. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[30] Karen Simonyan, Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Fisher vector faces in the wild. In *BMVC*, 2013.

[31] Michael Stark, Michael Goesele, and Bernt Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, 2010.

[32] Hao Su, Min Sun, Li Fei-Fei, and Silvio Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009.

[33] Min Sun, Hao Su, Silvio Savarese, and Li Fei-Fei. A multi-view probabilistic model for 3d object classes. In *CVPR*, 2009.

[34] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[35] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. 2014.

[36] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.

[37] M Zeeshan Zia, Michael Stark, Bernt Schiele, and Konrad Schindler. Detailed 3d representations for object recognition and modeling. *PAMI*, 35(11), 2013.