Jointly Optimizing 3D Model Fitting and Fine-Grained Classification

Yen-Liang Lin[1], Vlad I. Morariu[2], Winston Hsu[1], and Larry S. Davis[2]

[1] National Taiwan University, Taipei, Taiwan
[2] University of Maryland, College Park, MD, USA
yenliang@cmlab.csie.ntu.edu.tw, whsu@ntu.edu.tw,
{morariu,lsd}@umiacs.umd.edu

Abstract. 3D object modeling and fine-grained classification are often treated as separate tasks. We propose to optimize 3D model fitting and fine-grained classification jointly. Detailed 3D object representations encode more information (e.g., precise part locations and viewpoint) than traditional 2D-based approaches, and can therefore improve fine-grained classification performance. Meanwhile, the predicted class label can also improve 3D model fitting accuracy, e.g., by providing more detailed class-specific shape models. We evaluate our method on a new fine-grained 3D car dataset (FG3DCar), demonstrating our method outperforms several state-of-the-art approaches. Furthermore, we also conduct a series of analyses to explore the dependence between fine-grained classification performance and 3D models.

1 Introduction

Fine-grained recognition methods have been proposed to address different types of super-ordinate categories (e.g., birds [10,5,6,8], dogs [18] or cars [23,14]), and many of these methods focus on finding distinctive 2D parts for distinguishing different classes [6,27,5,2] or seeking better pose-invariant feature representations [29]. Recently, researchers [8,14] have used 3D models for fine-grained classification. While these methods have shown some success in tackling viewpoint variations within the objects, their non-deformable 3D model representations limit the ability of these approaches to adjust to different shapes of objects.

At the same time, 3D object modeling has also received renewed attention recently [20,12,21,16,30]. Many methods have been proposed to fit a 3D model to a 2D image [16,30,24]. However, their objective functions are usually highly non-linear and a suboptimal initialization leads to convergence to poor minima. One common approach is to try multiple starting points [30]; however, this increases the time to reach convergence and it is also unclear how many starting points are sufficient for good results.

In this paper, we investigate these two challenging problems together and show that they can provide benefit to each other if they are solved jointly. We propose a novel approach that optimizes 3D model fitting and fine-grained classification in a joint manner. 3D model representations can convey more information than

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 466–480, 2014.

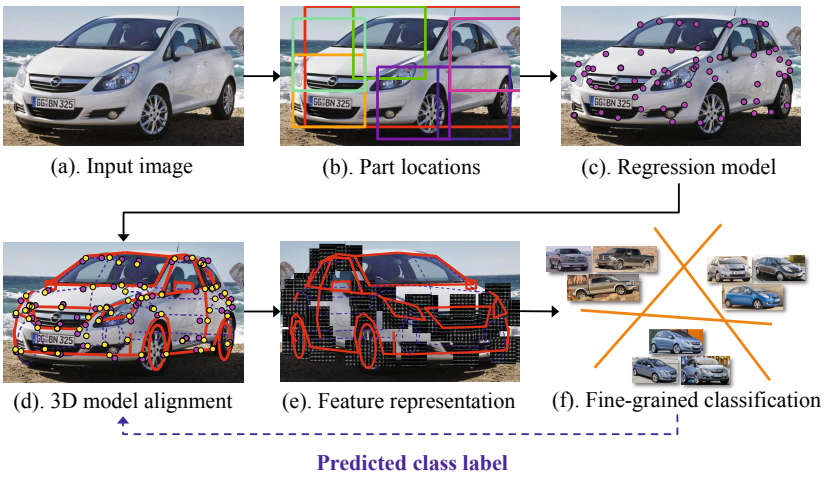


Fig. 1. System overview of the proposed method. Given an input image (a), our method first extracts rough part locations based on deformable part models (DPM) (b) and then uses regression models to estimate image landmark locations (c). Next, we fit the 3D model landmarks (yellow circles) of our 3D deformable model to the predicted 2D landmark locations (magenta circles) (d), extract part-based features relative to the 3D geometry (e) and feed these features into SVM classifiers for fine-grained classification (f). After classification, the predicted class labels are then further exploited to iteratively refine the model fitting results.

traditional 2D-based approaches, e.g., viewpoint and precise part locations, and can therefore benefit fine-grained classification. Also, the semantic label of each part is typically defined in modern CAD file formats, which reduces the naming effort by users [6]. Additionally, the predicted class label provides a better class-specific shape prior, which improves model fitting by alleviating the local minima problem for non-linear objective functions. Instead of using a rough 3D ellipsoid [8] or a massive bank of classifiers [14], we adopt a more general and flexible 3D modeling approach, which is based on a highly detailed and deformable 3D model constructed by Principal Components Analysis (PCA) on a set of 3D CAD models.

The system overview is depicted in Fig. 1. Given an input image (a), we first apply a deformable part model (DPM) [9] to obtain rough part locations (b) and feed them as features to a pre-trained regression model for estimating landmark locations (c). The 3D object geometry is recovered by fitting a deformable 3D model to those estimated 2D landmark locations (d). Then, we represent each image by the concatenation of feature descriptors (e.g., HOG [4] or Fisher vector [22]) for each landmark (e). SVM-based classifiers are utilized for fine-grained classification (f). Predicted classes are then exploited to derive better shape parameters to refine the 3D model fitting results in an iterative manner.

The main contributions of this work include:

- We simultaneously optimize 3D model fitting and fine-grained classification in a joint manner. As shown in our experimental results, they benefit each other and lead to improved performance on both tasks (see Tables. 1 and 3).
- We propose a general 3D model fitting approach, a landmark-based Jacobian system, for fine-grained classification; it is shown experimentally to outperform several state-of-the-art 2D-based approaches on a new fine-grained 3D car dataset (FG3DCar).
- We also provide an in-depth analysis of various design decisions to explore the dependence of 3D models and fine-grained classification.

2 Related Work

Fine-Grained Classification: Various methods have been proposed to find distinctive 2D parts. In [6] Duan et al. propose a latent conditional random field model that automatically discovers discriminative attributes. Yao et al. [27] select important regions by a random forest with discriminative decision trees. Deng et al. [5] introduce a human-in-the-loop approach to select discriminative bubbles. Gavves et al. [10] localize distinctive details by roughly aligning objects. Some researchers also seek better feature representations for pose invariance [29].

However, there is currently little research employing 3D models for fine-grained classification. Farrell et al. [8] fit an ellipsoid to 2D images of birds and use it to construct a pose-normalized feature representation. However, a rough ellipsoid might not be suitable for other categories (e.g., car). The most related work to ours is [14], which lifts 2D-based features into 3D space to better associate features across different viewpoints. However, they use a massive bank of classifiers (i.e., example-based) to match 3D models to 2D images, which is time consuming and not applicable to different object shapes.

3D Modeling: At the same time, there has been renewed attention in representing objects in 3D rather 2D [9,19,15]. 3D model representations can convey more information than traditional 2D-based approaches, such as viewpoint, precise part locations, model shape and semantic meaning of parts, and can benefit high-level object reasoning. There are some recent works tailoring 2D part-based methods (e.g., DPM [9]) toward 3D geometric reasoning [21,20], however these approaches only provide coarse bounding boxes in either 2D or 3D space. To go beyond a bounding box representation, Hejrati et al. [12] recover a coarse 3D model from 2D part locations with non-rigid SfM. Some recent works go even further by fitting a more detailed 3D model to 2D images [30,16,24]. However, the objective functions for these methods are usually highly nonlinear and often get trapped in local minima. One possible solution to this problem is by sampling [30], that is, having multiple starting points and selecting the best solution. However, this lengthens the time to convergence and it is still not clear how many starting points are needed for good results.

Inspired by some co-optimization approaches for other tasks [28,15,1], we combine 3D model fitting and fine-grained classification jointly and show that

they benefit each other. We propose a more general 3D modeling approach for fine-grained classification based on the Active Shape Model (ASM) formulation, which is more flexible and effective than using a large set of classifiers [14] or a rough ellipsoid [8]. Furthermore, we exploit classification results (i.e., class labels) to derive better shape priors for improving 3D model fitting accuracy. Both processes collaborate iteratively.

Algorithm 1. Overall algorithm

Input: Given an input image I

Output: Class label c^*, shape s^* and pose x^*.

1: Find part locations and component $(z, m) = DPM(I)$
2: Estimate image landmark locations $\hat{l} = regression(z, m)$ Eq. (2)
3: Initialize shape: $\mathbf{s}^{(0)} \leftarrow \mathbf{u}$ (mean shape) and pose: $\mathbf{x}^{(0)} \leftarrow \mathbf{x}_{init}$ parameters
4: **for** $t = 1$ to T **do**
5: $(\mathbf{s}^{(t)}, \mathbf{x}^{(t)}) \leftarrow FitModeltoImage(\mathbf{s}^{(t-1)}, \mathbf{x}^{(t-1)})$ Eq. (5) & Eq. (6)
6: $f(I) \leftarrow ExtractFeatureVector(\mathbf{s}^{(t)}, \mathbf{x}^{(t)})$ Eq. (11)
7: $c \leftarrow Classification(f(I))$
8: Refine shape parameters $\mathbf{s}^{(t)} \leftarrow \Phi(c)$
9: **end for**

3 3D Deformable Car Model

To cope with large shape variations, we build our 3D representation based on the Active Shape Model (ASM) formulation [3]. Each instance (3D model) is represented by a collection of 3D points. These points have the same semantic meaning (i.e., are located on the same car parts) across different 3D models. Then we perform PCA to derive mean $\mathbf{u}$ and n eigen-vectors $\Omega = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n]$.

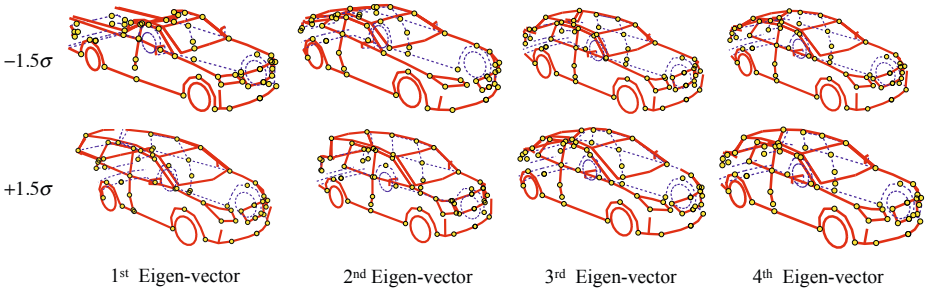


Fig. 2. The top four eigen-vectors derived from Active Shape Model (ASM) are visualized with the shape parameters $\pm 1.5\sigma$ (eigen-value), where landmarks are drawn as yellow circles and (hidden) model segments estimated from 3D geometry are drawn as (blue dotted) red lines.

Any 3D model can be represented as a linear combination of n eigen-vectors with shape parameters $\mathbf{s} = [s_1, ..., s_n]^T$:

$$P'(\mathbf{s}) = \mathbf{u} + \sum_{i=1}^{n} s_i \mathbf{w}_i \tag{1}$$

In our experiments, we use 11 3D CAD models of cars for training our 3D deformable model, including 3 sedans, 2 wagons, 1 pickup truck, 1 crossover, 2 hatchbacks, and 2 SUVs. There are total 256 salient points and 342 triangular faces; from them we manually select 64 landmarks covering important appearance and shape features for car images (see Fig. 2).

4 Regression Model for Landmark Estimation

To fit our 3D model to images, we locate the corresponding landmarks in the 2D image using a set of regression models based on part locations from DPM.

Our approach differs from the previous approaches that find the correspondences between image edges and model segments based on some low-level features (e.g., edge intensity) [16,24], which often fail due to cluttered background or complex edges on the surface of cars. Also, we avoid training a part detector for each landmark individually [17,30], which ignores the geometric relations between parts and may generate a noisy detection map with several local maxima. Instead, we exploit part locations generated from a state-of-the-art part-based method (e.g., [9]) to estimate the image landmark positions, which implicitly encodes both appearance of and geometric relationships between landmarks.

More formally, the input is a set of training images with detected part locations: $z = \{\beta_1, \beta_2, ...\beta_o\}$, where β_i denotes the pixel coordinates for the bounding box of each part (see Fig. 1 (b)), component number m from DPM, and manually annotated landmark positions $l = \{l_1, l_2, \cdots, l_N\}$, where l_i specifies a 2D position (x, y) for i-th landmark (see Sec. 6.1 for more details of obtaining ground truth landmark locations). We then train a regression model for each landmark under each component using the part locations as input features:

$$l_i \approx f(z) \tag{2}$$

We use linear Support Vector Regressor as our regression model to train each landmark position x, y separately. At test time, we use the pre-trained regression models to estimate image landmarks $\hat{l} = \{\hat{l}_1, \hat{l}_2, \cdots, \hat{l}_N\}$ given the part locations and component number from DPM. Example estimated landmarks are shown in Fig. 1 (c).

Given the mean car shape and initial pose, the goal of model fitting is to adjust the pose and shape parameters to minimize the distances between model and image landmarks. For initial pose, we roughly estimate the translation, rotation and scale from the DPM model [9]. For shape, the mean shape is adopted in the first iteration and can be further refined by exploiting the predicted class label. There do exist approaches for solving shape and pose parameters [16,30,24],

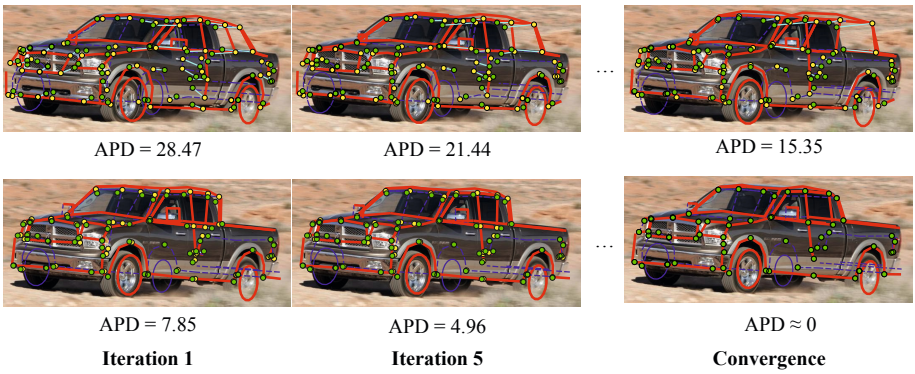


Fig. 3. Illustration of the local minimum problem when using an improper initial shape; ground truth pose and corresponding landmarks are used in this example. The model fitting accuracy is measured by average pixel distance (APD) (Eq. 12). **Top:** initialization by mean shape. **Bottom:** initialization by type shape (i.e., mean of all pickup truck shapes). Yellow circles are the landmarks on the model edges and green circles are the corresponding image landmarks from ground truth. This example illustrates that model fitting accuracy is strongly affected by the initial shape parameters due to the non-linearity of the target objective function. This motivates us to leverage the class label to obtain a better shape prior to improve model fitting accuracy.

however, their objective functions are usually non-linear and their fitting performance is sensitive to initialization. Additionally, the underlying shape distribution of cars is not a single normal distribution represented by a PCA model. There are several disjoint modes for different car classes and thus a generated car shape (controlled by shape parameters) might not be physically possible (e.g., the bottom example of 1st eigen-vector in Fig. 2).

Fig. 3 gives an example of 3D model fitting results when adopting different initial shape parameters; here ground-truth pose and landmark correspondences are used. Some car samples, e.g., pickup truck, are quite different from the mean shape, and optimizing shape starting from the mean shape is a very challenging even using ground-truth pose and landmark correspondences. To alleviate these problems, we use the predicted class label to refine our shape parameters, where the mapping function $\Phi(\cdot) : \mathbb{R}^1 \rightarrow \mathbb{R}^n$ that maps the class label to the shape parameters is learned in advance from our training samples (Sec. 6.1). To instantiate this idea, we modify the edge-based Jacobian system from [16] to landmark-based (Fig. 4), since edge-based approaches are susceptible to noise and clutter. Our method is general and could be also applied to other 3D model fitting approaches (e.g., [30]). For model fitting, the task can be formulated as minimizing an error function $F : \mathbb{R}^{n'} \rightarrow \mathbb{R}^N$ [16]:

$$\mathbf{q}^* = \arg\min_{\mathbf{q} \in Q} F(\mathbf{q}), \tag{3}$$

$$F(\mathbf{q}) = \mathbf{e} = (e_1, e_2, ...e_N), \tag{4}$$

which takes input parameter $\mathbf{q} = [s, x]^T \in \mathbb{R}^{n'}$ and generates N output errors, where s denotes shape parameters (n dimensions), x are pose parameters (3 rotation and 3 translation), Q is the parameter space. The total number of input parameters is $n' = n + 6$. The error vector $\mathbf{e}$ contains all error terms and each e_i denotes the signed distance error (i.e., the red line between u_i and v_i in Fig. 4 (b)) of the i-th model landmark to its corresponding image landmark. The solution can be obtained by iteratively solving a Jacobian system:

$$\mathbf{J}(\mathbf{q}_k)\Delta\mathbf{q} = -F(\mathbf{q}_k) = \mathbf{e}, \tag{5}$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \eta\Delta\mathbf{q}, \tag{6}$$

where $\mathbf{J}$ is the Jacobian matrix, and η is the learning rate (η is set to 0.1 in our experiments). To compute each Jacobian row $\mathbf{J}_i$ more easily, the error function can be split into to 3 composite functions and the Jacobian matrices can be computed by the chain rule:

$$e_i = F_i(\mathbf{q}) = F_i^1(F_i^2(F_i^3(\mathbf{q}))), \tag{7}$$

$$\mathbf{J}_i = \mathbf{J}_i^1\mathbf{J}_i^2\mathbf{J}_i^3, \tag{8}$$

where F_i^3 generates the corresponding 3D point, X_i, of landmark i from the input parameters $\mathbf{q}$; F_i^2 projects X_i into 2D image space u_i; and, F_i^1 measures the distance error between the projected landmark u_i and its corresponding image landmark v_i. We modify the distance error function and its Jacobian matrix to:

$$F_i^1(u_i) = n_i^T R(\theta)(v_i - u_i), \tag{9}$$

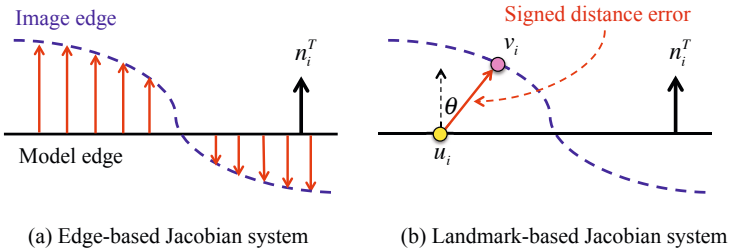
$$\mathbf{J}_i^1 = -n_i^T R(\theta), \tag{10}$$

where R is the rotation matrix and θ is the angle between $v_i - u_i$ and n_i^T (see Fig. 4). This modification enables the model to search for the most similar image landmark derived by our regression models (Sec. 4) without being constrained to the normal direction used in the original formulation. In other words, our model possesses the ability to match landmark-to-landmark instead of edge-to-edge; see [16] for more details.

5 Feature Representation for Classification

The appearance of cars in an image can change dramatically with respect to viewing angle and self-occlusion becomes an important issue for fine-grained categorization. We leverage the proprieties from 3D models to better deal with these problems.

To eliminate the need to model the direction that the car is facing in the image, we use the estimated pose from 3D model fitting to flip (mirror image) the car



(a) Edge-based Jacobian system (b) Landmark-based Jacobian system

Fig. 4. Comparison of edge-based [16] (a) and our landmark-based Jacobian system (b), where image and model edge are depicted in dotted blue and black line, u_i and v_i are the i-th model landmark and corresponding image landmark and n_i^T is the normal direction of the model edge. Our landmark-based Jacobian system finds the corresponding image landmarks by using regression models (encoding both appearance and geometric features) rather than low-level edge features (searching along the normal direction) as in the traditional edge-based approach. Therefore, our landmark-based Jacobian system is more efficient (only landmarks are needed for computing the Jacobian matrix) and robust to clutter and noise.

(for example, so that all cars point to the left of the image). Not surprisingly, flipping improves performance noticeably (Table. 2).

After flipping, we extract a feature descriptor φ_i from a window ($W \times W$, $W = 55$ in our experiments) centered around each landmark and concatenate them into a high-dimensional vector as our final feature representation:

$$f(I) = [v_1(\mathbf{q})\varphi_1, v_2(\mathbf{q})\varphi_2, ..., v_N(\mathbf{q})\varphi_N], \tag{11}$$

where $v_i(\mathbf{q})$ is a binary indicator function for visibility, which can be computed by normal direction of model faces. In other words, the final feature vector is modified by zero-filling the features corresponding to occluded landmarks as predicted by 3D geometry. Since those landmarks are self-occluded, their locations would be less stable compared to the visible ones and their features are less predictive of object class. The trimmed feature representation further boosts classification performance.

We explore two different feature descriptors: HOG [4] and Fisher vector [22], both of which are commonly used in classification. After feature extraction, we use a multi-class Linear SVM [7] to determine the class label.

6 Experiments

We present experiments to validate the effectiveness of our approach for fine-grained classification and 3D model fitting.

6.1 Fine-Grained 3D Car Dataset

Existing fine-grained car datasets (e.g., [13,23]) are not suitable for our purposes, since they are not annotated with both landmark locations and fine-grained

class labels. We created a new fine-grained 3D car dataset (FG3DCar) for this study[1], which consists of 300 images with 30 different car models under different viewing angles, e.g., sedan, SUV, crossover, hatchback, wagon and pickup truck. See examples in Fig. 7.

For each car image, we manually annotated 64 landmark locations. Instead of directly performing landmark annotation in the 2D image space, since it is difficult for humans to identify occluded landmark locations, we leverage the geometric constraints of 3D models to automatically infer the locations of occluded landmarks. We manually annotate the correspondences between visible 3D landmarks of our deformable 3D model and their 2D projections on the image, and iteratively adjust the shape and pose parameters to minimize the distance errors between the correspondences based on our modified Jacobian system. Our deformable 3D model is constructed from a set of 3D CAD models with manually aligned 3D points as discussed in Sec. 3. Our annotations provide not only the location and visibility state of each landmark but also the final shape parameters for each car instance.

We evenly split the images into a train/test set for evaluating classification performance. The mapping function $\Phi(\cdot) : \mathbb{R}^1 \rightarrow \mathbb{R}^n$ as mentioned in Sec. 4 is learnt by averaging the shape parameters within the same class from our training dataset. Foreground images are used, following standard criteria in fine-grained classification [23,14], and resized to height = 300 pixels. Note that the reported numbers in the following experiments (e.g., fitting accuracy) are based on this image scale.

6.2 Baselines

We compare our approaches with several state-of-the-art 2D-based methods: LLC [26], PHOW [25] and Fisher vector (FV) [22]. We only report the main parameter settings of the baseline methods here; for more details please refer to the original papers. For LLC, we train a codebook with 2048 entries and use 3-layer spatial pyramid (i.e., 1×1, 2×2 and 4×4). For FV, we reduce the dimensionality of SIFT feature to 64 by applying Principal Component Analysis (PCA) and use Gaussian Mixture Model (GMM) with different numbers of components ($e.g., K = 32, 64, 256$). Power- and L2 normalization schemes are also applied [22]. To roughly encode the spatial relationship of FV, we also combine FV with a [2x2] spatial pyramid. For both methods, we use linear SVM classifiers with the cost parameter $C = 10$. For PHOW, we train the same-sized codebook and 3-layer spatial pyramid as LLC and use a homogeneous kernel map for the χ^2 kernel. For our approach, we only use $K = 32$ components of FV on each landmark due to the high dimensionality of our final part-based feature representation.

To validate the effectiveness of these baseline methods, we apply them on a public fine-grained car dataset [23] (denoted as BMVC dataset). Experimental

[1] We will publicly release our dataset, landmark annotations, and source code at `www.cmlab.csie.ntu.edu.tw/~yenliang/FG3DCar/`

Table 1. Classification comparison. We report the results of our method and several state-of-the-art methods: LLC [26], PHOW [25] and Fisher vector (FV) [22] on BMVC [23] and a new fine-grained 3D car dataset (FG3DCar). The baseline methods show very competitive results on BMVC dataset compared to best reported methods [23,14] (not publicly available), demonstrating their effectiveness for fair comparison. We compare our method with these baselines on FG3DCar dataset (The reasons why we did not evaluate our method on BMVC dataset are explained in Sec. 6.2). Our 3D part-based representation shows superior performance compared to the baseline methods (shown in bold font), validating the feasibility of using 3D models to improve fine-grained classification performance. To further analyze where future work should focus, we also investigate classification performance under idealized cases (last two rows); the results show that better alignment (GT alignment) would lead to further improvements. See Sec. 6.3 for more detailed explanations.

Method	BMVC [23]	FG3DCar
LLC [26]	84.5%	51.3%
PHOW+χ^2 [25]	89.0%	54.7%
FV [22](K=32, 64, 256)	88.3%, 90.7%, 93.9%	62.0%, 64.7%, 70.0%
FV [22] [2x2](K=32, 64, 256)	90.9%, 91.7%, 92.6%	60.0%, 64.0%, 69.3%
structDPM [23]	93.5%	-
BB-3D-G [14]	94.5%	-
Regression + FV	-	82.7%
3D-part (mean prior)+(HOG/FV) **(ours)**	-	55.3% / **88.7%**
3D-part (class prior)+(HOG/FV) **(ours)**	-	57.3% / **90.0%**
3D-part+GT model prior+(HOG/FV)	-	70.0% / 90.0%
3D-part+GT alignment+(HOG/FV)	-	90.7% / 95.3%

results (left column in Table. 1) show that they achieve very competitive results compared to best reported methods [23,14][2]. Also, the classification performance on BMVC dataset is saturated, which is why we chose not to incur the cost of manual annotating 3D pose and did not evaluate our method on this dataset (since our regression models are currently trained based on manually annotated landmark locations). Instead, we will compare our methods with these baselines on our new and more challenging fine-grained 3D car dataset.

6.3 Fine-Grained Classification Results

We compare our 3D part-based representation to several 2D-based state-of-the-art approaches on our FG3DCar dataset. Empirically, we find that the convergence of our approach is achieved after 2 iterations (i.e., T in Alg. 1) for most cases when using Fisher vectors. Therefore, we only report the results for the first and second iteration, which are also denoted as "mean prior" and "class prior" respectively. In addition, we also provide an in-depth analysis of different

[2] The source codes of methods [23,14] are not publicly available.

choices of feature descriptors and 3D features, and also study the idealized cases for the task of fine-grained classification.

3D Versus 2D Representation. Table. 1 summarizes the overall classification accuracy for different methods. The overall performance of baseline methods on our dataset is lower than the BMVC dataset, implying that our dataset is more challenging as it contains more classes (i.e., 30 classes for ours versus 14 classes for BMVC). From the results, our 3D part-based representation (3D-part+mean/class prior+FV) significantly outperforms baseline methods, confirming the feasibility of using 3D models to improve fine-grained classification accuracy - they provide more precise part locations and tolerance to viewpoint changes. Moreover, the classification performance is further improved by using class prior (see mean prior vs. class prior in Table. 1), as it more closely matches to the instance shape, which supports the proposed iterative approach for further improvements.

In Table. 2, we investigate the impact of using 3D features (e.g., flipping and visibility). Flipping improves over un-flipped by 10% and visibility modeling further improves the results by 3%. It is worth noting that even when we do not use flipping and visibility, our method still outperforms baseline methods, gaining from the precise landmark locations derived from 3D models[3].

Fisher Vector Versus HOG. In Table. 1, we observe that the Fisher vector significantly outperforms HOG. We hypothesize that this is because Fisher vector adopts a bag-of-visual-words (BOW)-like feature representation, which ignores spatial relationships and thus can tolerate a higher amount of local displacement. We also find that the classification accuracy of HOG significantly improved (55.3% to 70.0%) when using ground truth model prior (3D-part+GT model prior+HOG) (i.e., perfect shape parameters), indicating that HOG needs more accurate alignment to obtain good classification accuracy compared to the classification-oriented Fisher vector. To better understand the effectiveness of HOG and Fisher vector, we further investigate the classification accuracy of these two features under different degrees of misalignment. To do this, we generate test data by adding Gaussian noise to the ground truth (e.g., translation), where the degree of misalignment is quantified by mean average pixel distance (mean APD):

$$\frac{1}{K}\sum_{i=1}^{K} APD_i,\ APD = \frac{1}{N}\sum_{j=1}^{N} dist(m_j, g_j) \tag{12}$$

Here, m_j and g_j correspond to j-th landmark on the fitted model and ground truth. K is the number of testing images and N is the number of landmarks. Fig. 5 plots the classification accuracy versus mean average pixel distance using

[3] Geometric constraints (e.g., shape, visibility) from 3D models further improved the results from regression model. Based on our 3D model representation, we believe more sophisticated image rectification techniques (e.g., [11]) can be utilized for further improvements, but leave these for future work.

different features. The result further confirms that Fisher vector is less sensitive to misalignment than HOG.

Idealized Case. To understand where future work should focus, we also evaluate our model with idealized perfect shape parameters (GT model prior) and landmark alignment (GT alignment) from ground truth data. We find that the model prior does not improve performance, due to the imperfect image landmark locations estimated by DPM and our regression model. Using ground truth alignment, the performance is increased to 95%, suggesting the possible future benefit from improving landmark estimation accuracy. We will discuss this issue in Sec. 6.4.

Table 2. Different settings of 3D features are analyzed for the method: 3D-part+class prior+FV. The results show that using flipping and visibility state further improve the classification accuracy. Even if we do not use flipping and visibility, our method still outperforms baseline methods (77.3% vs. 70%), gaining from the precise landmark locations derived from 3D models.

Flipping	Visibility	Classification accuracy
No	No	77.3%
Yes	No	87.3%
Yes	Yes	90.0%

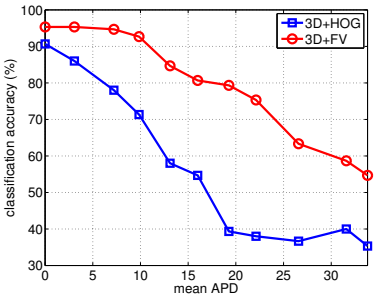


Fig. 5. Comparison of Fisher vector and HOG under different levels of misalignment. Fisher vector can tolerate more displacement error than HOG.

6.4 3D Model Fitting Results

Having discussed the power of our 3D part-based representation for fine-grained classification, we now describe experiments to evaluate the model fitting accuracy. There are two main sources of error for model fitting: initial parameters and estimated landmarks. We analyze their effects in the following paragraphs.

Class Versus Mean Shape Prior. Table. 3 shows the model fitting results, where "pose" indicates optimizing pose parameters only (keeping shape parameters fixed), while "pose+shape" optimizes both shape and pose parameters. Our results show that the class shape prior outperforms the mean shape prior, as it more closely matches the instance shape than the mean shape prior for highly non-linear objective functions as mentioned in Sec. 4. In Fig. 6, we further investigate this by evaluating the mean APD for each category. We see the largest improvement for those categories (e.g., pickup truck) that deviate the most from the mean shape, validating the utility of using the class label to improve the 3D model fitting. We also evaluate the edge-based Jacobian [16]; it obtains lower model fitting accuracy (e.g., mean APD = 43.6 and 45.0 for mean prior+pose and mean prior+pose+shape) than our landmark-based Jacobian, since it is susceptible to noise and clutter. Fig. 7 shows some fitting results of our system.

Pose Versus Pose + Shape. We find that optimizing pose alone yields better results than optimizing both pose and shape for the class prior case. A possible reason is that imperfect landmark positions estimated from DPM and our regression models introduce errors into the shape model parameters. Therefore, optimizing pose alone allows the shape model slightly compensate prediction errors from regression models and lead to better fitting results. Meanwhile, if

Table 3. Model fitting accuracy with different shape priors. Class prior achieves better fitting accuracy than mean prior, validating the effectiveness of using the predicted class labels to refine model fitting.

Method	mean APD
Initial parameter	44.4
Mean prior+pose	20.4
Mean prior+pose+shape	20.3
Class prior+pose	**18.1**
Class prior+pose+shape	18.8

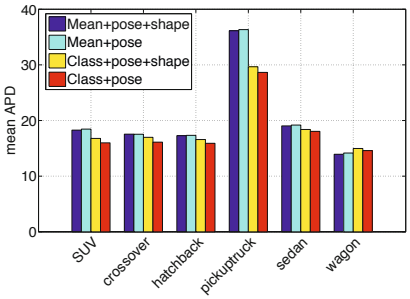


Fig. 6. Mean APD for each category. Class prior provides more benefit to the model fitting accuracy for those categories (e.g., pickup truck) that deviates far from the mean shape.

Fig. 7. Comparison of model fitting results. Top row shows the output of mean prior+pose+shape, and the bottom shows our final fitting results: class prior+pose. Our approach can produce better fitting results (e.g., the backside of pickup truck, SUV, hatchback and the grille part of Alfa romeo) compared to the mean prior, because it can more closely match to the target shape (benefiting from the predicted class label) and avoids falling into local minima for our non-linear objective function.

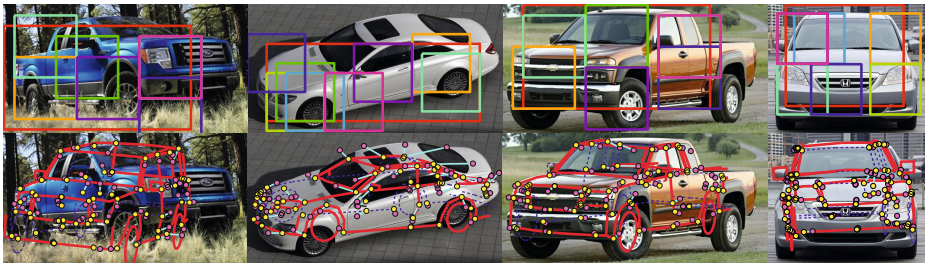


Fig. 8. Some failure cases of our system mainly caused by wrongly estimated part locations from DPM and errors introduced by regression models

the class label can be estimated correctly, the role of shape optimization might not be as important. We conjecture that the fitting performance can be further improved if we have more training images for each category so that we can train better regression models.

7 Conclusions and Future Work

In this work, we have presented an iterative approach for simultaneously optimizing 3D model fitting and fine-grained classification. By leveraging 3D models, we improved fine-grained classification performance over several state-of-the-art 2D-based methods, confirming the ability of our model to deliver more informative features than previous work. At the same time, we also showed that the predicted class label can further improve the 3D model fitting results. In future work, we seek further improvements on landmark estimation accuracy by using class label information and incorporate image rectification techniques (e.g., [11]) to better associate images across different viewpoints.

References

1. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: CVPR (2008)
2. Berg, T., Belhumeur, P.N.: Poof: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In: CVPR (2013)
3. Cootes, J.G.T.F., Taylor, C.J., Cooper, D.H.: Active shape models—their training and application. In: CVIU (1995)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
5. Deng, J., Krause, J., Fei-Fei, L.: Fine-grained crowdsourcing for fine-grained recognition. In: CVPR (2013)
6. Duan, K., Parikh, D., Crandall, D., Grauman, K.: Discovering localized attributes for fine-grained recognition. In: CVPR (2012)
7. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)

8. Farrell, R., Oza, O., Zhang, N., Morariu, V.I., Darrell, T., Davis, L.S.: Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In: ICCV (2011)
9. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan., D.: Object detection with discriminatively trained part based models. TPAMI (2010)
10. Gavves, E., Fernando, B., Snoek, C.G.M., Smeulders, A.W.M., Tuytelaars, T.: Fine-grained categorization by alignments. In: ICCV (2013)
11. Guo, Y., Rao, C., Samarasekera, S., Kim, J., Kumar, R., Sawhney, H.: Matching vehicles under large pose transformations using approximate 3d models and piecewise mrf model. In: CVPR (2009)
12. Hejrati, M., Ramanan, D.: Analyzing 3d objects in cluttered images. In: NIPS (2012)
13. Krause, J., Deng, J., Stark, M., Fei-Fei, L.: Collecting a large-scale dataset of fine-grained cars. In: CVPR-FGCV2 (2013)
14. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: International IEEE Workshop on 3D Representation and Recognition (2013)
15. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV (2007)
16. Leotta, M.J., Mundy, J.L.: Vehicle surveillance with a generic, adaptive, 3d vehicle model. TPAMI (2011)
17. Li, Y., Gu, L., Kanade, T.: Robustly aligning a shape model and its application to car alignment of unknown pose. TPAMI (2011)
18. Liu, J., Kanazawa, A., Jacobs, D., Belhumeur, P.: Dog breed classification using part localization. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 172–185. Springer, Heidelberg (2012)
19. Özuysal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: CVPR (2009)
20. Pepik, B., Gehler, P., Stark, M., Schiele, B.: 3d2pm - 3d deformable part models. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 356–370. Springer, Heidelberg (2012)
21. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3d geometry to deformable part models. In: CVPR (2012)
22. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
23. Stark, M., Krause, J., Pepik, B., Meger, D., Little, J.J., Schiele, B., Koller, D.: Fine-grained categorization for 3d scene understanding. In: BMVC (2012)
24. Tsin, Y., Genc, Y., Ramesh, V.: Explicit 3d modeling for vehicle monitoring in non-overlapping cameras. In: AVSS (2009)
25. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), `http://www.vlfeat.org/`
26. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR (2010)
27. Yao, B., Khosla, A., Fei-Fei, L.: Combining randomization and discrimination for fine-grained image categorization. In: CVPR (2011)
28. Chai, Y., Lempitsky, V., Zisserman, A.: Symbiotic segmentation and part localization for fine-grained categorization. In: ICCV (2013)
29. Zhang, N., Farrell, R., Darrell, T.: Pose pooling kernels for sub-category recognition. In: CVPR (2012)
30. Zia, M.Z., Stark, M., Schiele, B., Schindler, K.: Detailed 3d representations for object recognition and modeling. PAMI (2013)