Learning Features and Parts for Fine-Grained Recognition

(Invited Paper)

Jonathan Krause*, Timnit Gebru*, Jia Deng †, Li-Jia Li ‡, Li Fei-Fei*
* Stanford University: {jkrause, tgebru, feifeili}@cs.stanford.edu
† University of Michigan: jiadeng@umich.edu
‡ Yahoo! Research: lijiali@yahoo-inc.com

***Abstract*—This paper addresses the problem of fine-grained recognition: recognizing subordinate categories such as bird species, car models, or dog breeds. We focus on two major challenges: learning expressive appearance descriptors and localizing discriminative parts. To this end, we propose an object representation that detects important parts and describes fine-grained appearances. The part detectors are learned in a fully unsupervised manner, based on the insight that images with similar poses can be automatically discovered for fine-grained classes in the same domain. The appearance descriptors are learned using a convolutional neural network. Our approach requires only image level class labels, without any use of part annotations or segmentation masks, which may be costly to obtain. We show experimentally that combining these two insights is an effective strategy for fine-grained recognition.**

I. Introduction

Fine-grained recognition [1]–[7] refers to the task of distinguishing sub-ordinate categories such as bird species [8], [9], dog breeds [10], aircraft [11], or car models [12], [13]. It is one of the cornerstones of object recognition due to the potential to make computers rival human experts in visual understanding.

However, two major challenges need to be solved before fine-grained recognition can achieve this goal. First, recognizing fine-grained classes typically requires differentiating fine details in appearance. It calls for an appearance representation that retains details critical for discrimination and discards unnecessary information. The retained discriminative details can be very subtle and highly domain-specific. For example, the two cars in Fig. 1 differ at the front bumper and turning signal. Descriptors such as SIFT [14] or HOG [15], while successful in more coarse-grained recognition tasks, may not be discriminative enough to differentiate at this level of detail.

Another challenge is in discovering and locating the parts that contain discriminative details. When humans describe differences between fine-grained classes, we almost always point out the location ("vertical bars on the car's grille", "black patch on the bird's beak"). That is, we locate the relevant object parts and then check the appearance. The Beetles in Fig. 1 are much easier to differentiate when told to explicitly look at the front bumper. This brings forward the issue of part discovery – which parts are discriminative and where are they? One possibility is to annotate the location of various parts by hand. However, this approach is costly and it may be difficult to scale up to handle many different types of fine-grained classes. We hypothesize that the ultimate solution to fine-grained recognition should entail both localizing important parts with minimal supervision and effectively describing their appearances in a way that does not discard information useful for classification.

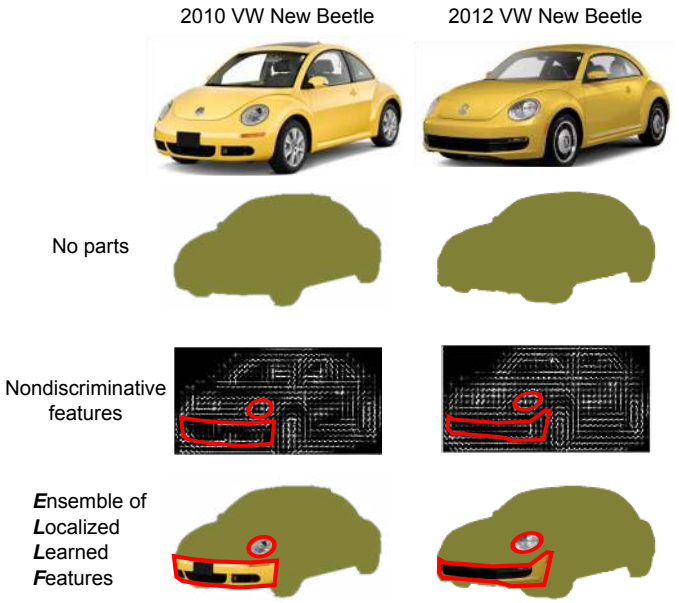

Fig. 1. The key to fine-grained recognition is localizing important parts and representing part appearance discriminatively, as global cues describing the overall shape or color often cannot capture the subtle differences. Without any information at the level of parts it is difficult to differentiate between two very similar classes. Similarly, features such as HOG [15], which are not discriminative for fine-grained classes, do not contain enough information.

In this paper we simultaneously tackle both feature learning and part discovery. Our central idea is *learning* both features and parts to form a unified object representation. Specifically, we use convolutional neural networks (CNNs) to learn appearance descriptors, and perform unsupervised part discovery to obtain a collection of part detectors. By learning the features appropriate to describe the object categories in question, we let the data determine which features are effective for discrimination, which helps avoid losing information useful for categorization. By keeping part discovery completely unsupervised with respect to part annotations, we aim to make our algorithm scalable to a variety of fine-grained domains, including ones for which it is not known a priori which parts are discriminative. In recognition time, we detect parts and

represent their appearances using the learned features, leading to an "Ensemble of Localized Learned Features" (ELLF), a novel representation for fine-grained recognition. To our knowledge our approach is the first to integrate feature learning and unsupervised part discovery in fine-grained recognition.

II. Related Work

A. Part-based representations

Many fine-grained recognition techniques involve part-based representations inspired by work on generic object recognition [16], [17]. Some explicitly model pose [2], [18] whereas others use less structured approaches [4], [19]–[21]. Typically part detectors are learned using hand-annotated key-points. Our approach departs from most prior work in that the part detectors are learned with zero supervision, which means that we can tackle multiple domains, including ones for which nothing more than class labels and bounding boxes are available.

B. Feature Learning

Feature learning is a promising approach that can generate powerful appearance representations. Much work has focused on encoding low-level features such as SIFT or HOG (e.g. [5], [22], [23]) or mining discriminative templates [5], [6]. The recent success of convolutional neural networks [24], [25] on large-scale classification and face recognition [26], [27] demonstrates that powerful features can be learned directly from pixels. This inspires us to adopt convolutional neural networks (CNNs) for fine-grained recognition. Note that unlike the DeCAF system [28] that trains features on ImageNet [29], we do not perform any pre-training using additional data. This requires care when choosing the network architecture and necessitates using a larger variety of data deformations in order to cope with and increase the size of the training set. To our knowledge this is the first time deep neural networks have been used for fine-grained recognition without any form of domain adaptation.

C. Other approaches

In addition to the approaches outlined above, segmentation has also been found to be particularly useful [21], [30]–[32] in fine-grained recognition tasks. Another line of research focuses on putting humans in the loop [33], [33]–[36]. These are complementary approaches that can be jointly used with our method, and we do not attempt to incorporate these additional cues in our work.

III. Approach

A. Overview

Our representation builds on the intuition that we need to localize parts and then compare their appearances. Fig. 2 provides an overview of the algorithm. The main idea is to have a representation that enables easy comparison of appearance features on corresponding parts. This leads to ELLF: Ensemble of Localized Learned Features. Suppose we have a collection of n object parts with associated part detectors, which we assume for now have already been trained. Given an input image (Fig. 2(a)), let a_i be the appearance of part i, as described by a convolutional neural network (Fig. 2(b)). The ELLF representation is then simply $(a_1, a_2, \ldots, a_n)$, the concatenation of part appearances (Fig. 2(c)). Note that due to view point change and occlusion, not all parts are necessarily detected. When part i is not detected, the appearance a_i is set to zero, preventing a classifier (Fig. 2(e)) from using any information at that part. With images represented by ELLF, we can then train classifiers such as linear SVMs to perform fine-grained classification. For us, the collection of parts is determined in an unsupervised framework and they are described using features from a convolutional neural network.

One desirable property of using ELLF is that it compares the appearances of each part and aggregates the similarities together. This is different from traditional approaches for generic object recognition such as spatial pyramid matching (SPM) [37] where a linear kernel compares the appearances at the same *spatial location* instead of the same *part*. SPM is thus sub-optimal for objects of different poses, because all the parts are not necessarily visible or at the same location across images.

We would like to highlight one other difference between ELLF and traditional bag-of-words representations. A linear kernel defined on bag-of-words histograms or its softly quantized generalizations such as LLC [22] roughly corresponds to comparing the *presence* of each visual word. In this case we have already quantized the appearances into visual words and are only checking whether specific visual words occur. The subtle appearance differences for fine-grained classes might still get lost in quantization. In contrast, since we describe object parts using features trained from a CNN, our representation keeps rich appearance descriptors in the final representation.

Now that we have defined ELLF, we proceed to describe the process of generating ELLF. There are two key components: learning discriminative appearance features and discovering parts.

B. Feature Learning

A hallmark of fine-grained recognition is that it demands rich and expressive appearance descriptors, as traditional descriptors like SIFT [14] or HOG [15] may not capture the right balance between discriminativeness and invariance for fine-grained classes. To this end we adopt the philosophy of end-to-end training of feature descriptors using neural networks, allowing the descriptors to adapt to the idiosyncrasies of individual categories. To our knowledge this is the first time deep feature learning is applied on fine-grained recognition involving no pre-training with additional data. We demonstrate that even on relatively small datasets, feature learning can be effective for fine-grained recognition.

In particular, we use a convolutional neural network (CNN) [24] that accepts pixels as its input and outputs probabilities of classes. We modify the architecture of Krizhevsky et al. [25] to account for our smaller-scale data, which we have found to be very important for preventing overfitting. The network consists of two convolutional layers followed by three fully connected layers with a softmax loss. Each convolutional layer performs convolutions with a bank of filters on the 3D input matrix and outputs filter responses in the

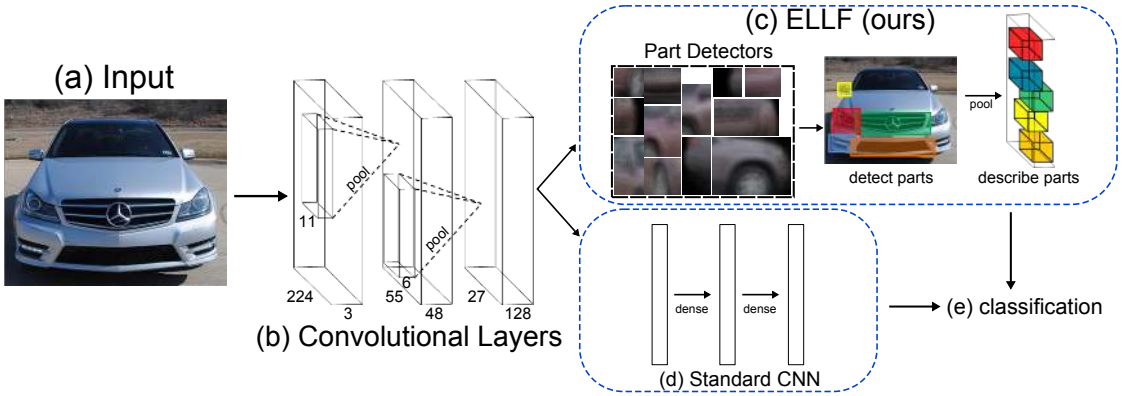

Fig. 2. Overview of our Ensemble of Localized Learned Features (ELLF) representation. Given an input image (a), we detect parts using a collection of unsupervised part detectors (Sec. III-C). We also feed the image into a convolutional neural network (CNN) (b) that outputs a grid of discriminative features (Sec. III-B). The CNN is learned with class labels and then truncated, retaining the first two convolutional layers which retain spatial information. We describe the appearance of each detected part using the learned CNN features by pooling in the detected region of each part (c). Appearance of of any undetected parts is set to zero. This results in our ELLF representation that is then used to predict fine-grained object categories (e). In comparison, a standard CNN (d) passes the output of the convolutional layers through several fully connected layers in order to make a prediction.

form of a 3D matrix. Since filter parameters are learned from the data, the network has the potential to generate feature descriptors tailored to specific domains. More details are given in Sec. IV-A.

After training, we remove the fully connected layers and use the two convolutional layers as a generator of pixel-level appearance descriptors. Note that it is necessary to cut off the features at this point in order to maintain spatial information – features in the fully connected layers are completely unordered. To obtain a descriptor for a region (such as a bounding box given by a part detector), we perform max-pooling of the descriptors located inside the region. Thus, one way to interpret our parts is as movable pooling regions in a CNN architecture.

C. Part Discovery

The goal of part discovery is to obtain a collection of reliable part detectors. Our key contribution is a part discovery algorithm that is fully unsupervised. Previous work has relied on hand-annotated keypoints to train part detectors [4]. Here we bypass human annotations completely, which has the advantage of scaling to very large-scale datasets.

How can we train part detectors without any annotations? The key observation is that objects with the same pose can often be automatically discovered by local low-level cues. Aligning poses between images is, in general, a difficult problem, because appearance may vary wildly even within the same category. However, localizing parts primarily depends on an understanding of the overall object shape without the need to scrutinize the local details—a blurred image of a dog may prevent you from recognizing the breed but will likely hold enough information for you to localize the parts.

This intuition motivates our part discovery procedure. We first discover sets of aligned images with similar poses. Under the assumption that images within a set are well aligned, the same parts have similar locations across images. We can thus train a part detector using the patches from the same spatial location as positive examples and patches from elsewhere as negative examples. Fig. 3 (top) illustrates this intuition. We now elaborate on the individual steps.

1) Discovering Aligned Images: The first step is discovering sets of aligned images. We use a randomized algorithm. We pick a seed image at random (Fig. 3(a)) and then retrieve nearest neighbors in terms of HOG features, extracted at multiple scales. To help reduce the influence of the background, we perform GrabCut [38] before extracting HOG features, initializing the foreground model with the object's bounding box, which is typically given in fine-grained recognition. These foreground segmentations are centered for the purpose of comparing across images. We repeat this process, randomly sampling multiple sets and using each set to generate multiple part detectors. With a reasonable number of training images to choose from, this method typically results in a set of images with nearly the same pose (Fig. 3(b)).

2) Part Selection: Next we select the parts to detect, as every location within the segmented foreground can be a potential part. To address this issue, we randomly sample a large number of regions with various sizes as candidates (Fig. 3(c)). We then select the parts with the highest energy, as measured by the variance of HOG across images (Fig. 3(d)). This helps prevent selecting parts which lack discriminative information – a part which does not vary at all across images cannot be useful for discrimination. Each time we select a part, we remove from the candidate list any parts that overlap more than a fixed threshold ρ with an already selected part, set to 15% in our implementation. This helps prevent learning redundant parts for a given set of aligned images.

3) Detector Learning: We then learn a detector for each selected part (Fig. 3(e)). Specifically, let I_j be the aligned images and z^+ be location of the selected part. Under the assumption that the images are well aligned, our learning objective for the part detector is finding a template w that minimizes the hinge loss

$$\min_w \quad \begin{array}{c} \sum_j \max\{0, 1 - w^T h(I_j, z^+) \\ + \sum_j \sum_{z_j^-} \max\{0, 1 + w^T h(I_j, z_j^-)\}, \end{array} \tag{1}$$

where $h(I_j, z^+)$ extracts features (HOG) on image I_j at positive patch location z^+. The variable z_j^- are the locations

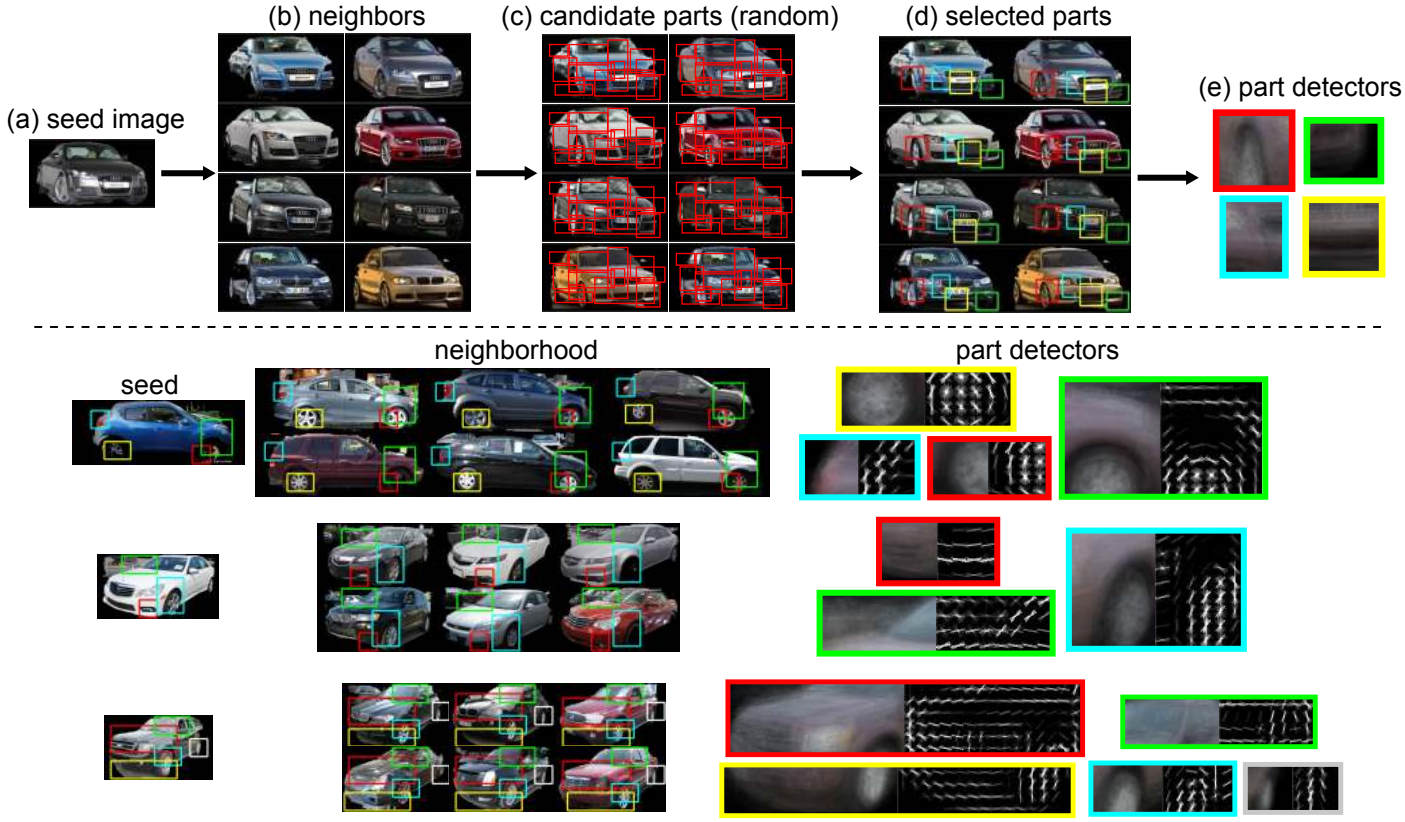

Fig. 3. Top: Our fully unsupervised part discovery pipeline. We randomly sample a seed training image (a) and retrieve the nearest neighbors (b) in terms of global HOG appearance. This allows us to identify well aligned images with similar poses. From a large sample of random parts (c) we pick the sub-windows with high energy (d) as candidate parts, which are then used to train our final part detectors (e), visualized as the average of the patches used as positive examples in training. Bottom: Examples of parts discovered by our method. Leftmost is the seed image used to generate the set of aligned images, a subset of which is shown in the middle. Shown at the right are the average of the image patches used as positives to train each part detector and the learned weights. Our method is able to discover a variety of parts from each neighborhood.

of the negative patches on image I_j, chosen randomly such that they do not overlap with the positive patch at location z^+.

We now relax the assumption that the images are well aligned to be robust to misalignment. Instead of having a fixed location z^+, we introduce a latent variable z_j^+ to represent the true location of the part on image z_j^+. Our learning objective is thus

$$\begin{aligned} \min_w \quad & \textstyle\sum_j \max\{0, 1 - \max_{z_j^+} w^T h(I_j, z_j^+) \\ & + \textstyle\sum_j \sum_{z_j^-} \max\{0, 1 + w^T h(I_j, z_j^-)\}, \end{aligned} \tag{2}$$

where we search for the best match over all possible locations z_j^+. The objective can be optimized by alternating between optimizing z_j^+ with fixed w and optimizing w with fixed z_j^+, similar to the latent SVM optimization introduced in [17]. We initialize the latent variable z_j^+ with the original location z^+. Also similar to [17], we augment the HOG feature $h(I, z)$ with $(dx \cdot dy, dx^2, dy^2)$ to include a spatial prior that penalizes patches too far away from the original z^+. Here $dx = x_z - x_{z^+}$ and $dy = y_z - y_{z^+}$, where (x_z, y_z) is the coordinate of the location z and (x_{z^+}, y_{z^+}) is the coordinate of the original location. This effectively defines a Gaussian prior on the true location relative to the original location z^+, preventing part detectors from spuriously firing at regions that by chance appear similar to the part while still allowing the parts themselves to move around in order to best fit the actual part location in each image.

At detection time, we set a threshold τ on the detector response. If the response is below τ, the part is considered not visible in the image and its appearance descriptor will be set to zero, preventing the classifier from receiving any information about a part that is not present.

4) Ensemble of Parts: To obtain a collection of part detectors, we repeat our discovery procedure multiple times. It is worth noting that the randomization throughout our discovery procedure can help increase the robustness of the recognition algorithm. As we will demonstrate in our experiments, increasing the number of randomly sampled part detectors improves performance. See Fig. 3 (bottom) for more examples of our part discovery pipeline.

IV. Experiments

A. Datasets and Implementation Details

We evaluate our algorithm on the Cars [13] fine-grained benchmark. We follow the standard evaluation procedure: all training and testing is performed on cropped images from the object bounding boxes. We report classification accuracy, i.e. the accuracy as averaged over the test examples. In our

experiments no annotations except class labels and bounding boxes are used in training.

We use the `cuda-convnet` implementation [25] of CNNs. The network consists of 48 11×11 filters with 3×3 pooling regions with stride 2 for the first convolutional layer, and 128 6×6 filters with pooling regions of size 6×6 every 6 pixels for the second convolutional layer. The number of units for the three fully connected layers are all 2048. All units are rectified linear units except for the fully connected layers, where we found that linear units work best. We resize each image to 256×256 and use various techniques for CNNs to prevent overfitting, including dropout, color perturbation, random rotations, and sampling subwindows of 224×224, as described in [25]. This architecture was determined by extensive experiments using a validation set drawn from the training set, as the network used in [25] suffers from substantial overfitting on fine-grained datasets, which typically have over 100x less training data than in ILSVRC2012 [39], the dataset used to train [25]. At test time, we average predictions over the four corners, middle patch, and their horizontal flips.

For part discovery, each aligned set consists of 1 query image and 49 nearest neighbors. To generate the candidate parts from the aligned images, we randomly sample 5000 patches and pick the top 10 with the highest HOG energy, measured across images. The threshold τ used for part detection is set to -1. While this low threshold results in part detections in nearly every image, even ones in which the part is not present, we have found that this improves performance, with the intuition that the small amount of signal we get by increasing the number of part detections is worth the corresponding increase in noise.

To train our final classifier, a linear SVM, we use as data ELLF features extracted on the original images as well as their horizontal flips. Note that this means that the classifier itself does not have access to some of the data deformations – namely, the color perturbations, random rotations, or subwindow sampling. It also does not use dropout for regularization. Although in principle one could train the SVM with these deformations, part detection remains a relatively costly operation compared to extracting CNN features, which makes computing ELLF features on many deformations expensive. At test time we average predictions over each test image and its horizontal flip to produce the final classification.

B. Results and Analysis

Table I reports our results compared with prior work. ELLF beats the previous state of the art, LLC [22], as well as BB [34] and BB-3D-G [13], two works designed for fine-grained recognition. This validates the claim that learning discriminative features and using them with parts discovered automatically is an effective strategy for fine-grained recognition. In the following sections we present more analysis.

1) Feature Learning: A plain CNN (70.5%) already outperforms previous work using traditional features such as SIFT or HOG (BB, BB-3D-G, and LLC). This is without extra unlabeled data or any kind of pre-training. It suggests that feature learning is able to generate rich appearance descriptors that adapt to particular categories, even with our limited amount of data.

Method	Accuracy
BB [34]	63.6
BB-3D-G [13]	67.6
LLC [22]	69.5
CNN-SPM (small)	67.9
CNN-SPM (large)	69.3
CNN	70.5
ELLF (ours)	73.9

TABLE I. MAIN RESULTS ON CLASSIFYING CARS. BB, BB-3D-G, AND LLC RESULTS AS REPORTED IN [13].

2) Usefulness of Parts: We next perform a control experiment that showcase the key benefits our ELLF representation—the same segments of the representations from two images refer to appearances of the same part. To verify this intuition, we replace our detected parts with SPM grids, where the same segments of the representations refer to the same image location instead of part. CNN-SPM(small) and CNN-SPM(large) in table I report the results of this control experiments. CNN-SPM(small) uses 1x1, 2x2, and 4x4 spatial pooling regions, and CNN-SPM(large) in addition uses 8x8 regions, both building on top of our CNN features. The ELLF representation outperforms both standard and very high-dimensional SPM representations, demonstrating that it is indeed helpful to enable comparing appearances at corresponding parts and that the gains from using ELLF are not simply due to feature dimension. Note that the performance of both CNN-SPM(small) and CNN-SPM(large) is below that of a standard CNN, which is due to the lower number of deformations the SVM classifier is trained on and the fact that it is not trained using dropout (see Sec. IV-A for details).

3) Consistency of Parts: To validate that the discovered parts generalize beyond our training data, we show a sample of parts and their top ten detections on the test set in Fig. 7. The top two rows show that discovered parts tend to fire rather consistently, even under mild changes in viewpoint. Examples of failure cases are given in the last row, in which 180-degree rotations of cars cause the part detectors to misfire on patches which, although locally very similar to the target part in appearance and position, are nonetheless different parts.

4) Number of Parts: We also investigate how much part discovery contributes to performance. Fig. 4 plots the classification accuracy versus number of part detectors discovered. It also plots the performance of directly using full CNN models without part discovery (CNN). Performance increases with the number of parts, up to a point when it plateaus. Remarkably 100 part detectors are sufficient to significantly improve the standalone CNN model. This shows that part discovery is an essential component of our representation. Eventually performance saturates (at around 1000 parts discovered).

5) ELLF vs. CNN predictions: In Fig. 5 we show a sampling of images where our method was correct and a standard CNN was incorrect, with the parts that contributed most toward the decision value of the correct class displayed. Our part detectors fire on a diverse range of parts, whereas a CNN is confined to a fixed pooling grid. In Fig. 6 we show example failure modes of ELLF, where the CNN was correct but ELLF was not. One disadvantage of ELLF's reliance on GrabCut for segmentation is that part detection suffers

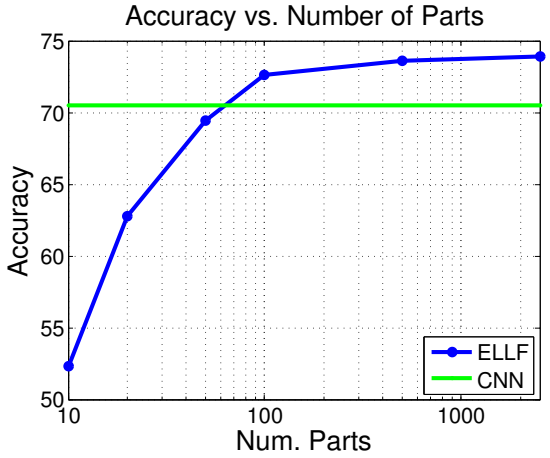

Fig. 4. Car classification accuracy versus the number of parts used in our ELLF representation. Results achieved using CNN alone are also included (green line).

when the segmentation is flawed, hurting our recognition performance.

6) Confusing Classes: In Fig. 8 we show the pairs of classes most confused with one another. The confusion score between a pair of classes C, D is determined by

$$Conf(C, D) = \frac{P_{C,D}}{\sum_i P_{C,i}} + \frac{P_{D,C}}{\sum_i P_{D,i}} \quad (3)$$

where $P_{a,b}$ is the number of images with ground truth label a predicted as class b. These pairs tend to consist of different models or years within the same make, except for the classes Chevrolet Express Cargo Van 2007 vs. GMC Savana Van 2012, in which case the difference in make is visually represented only by the logo small on the front of the van.

7) Most Useful Parts: Which parts, discovered in an unsupervised fashion, are most useful for discrimination across all 196 categories? To measure this, we sum the absolute value of the learned classifier weights across classes and dimensions for each part to produce an importance score for each part. The top 10 parts are shown in Fig. 9. These parts are relatively large and thus can give information about the overall shape or type of car, e.g. whether the car is a sedan, SUV, or coupe. We also observe that these parts tend to occur in the top half of the automobiles, almost never overlapping with the tires. This agrees with the intuition that tires are not useful regions to look at when discriminating cars.

C. Limitations

Although it is a step in the right direction, ELLF is far from perfect. Performance can be improved by pre-training the CNN on ImageNet [25], [29], especially for smaller datasets like the Cars dataset [13] used in this paper. However, such constraints will be alleviated as fine-grained datasets continue to grow in size. Second, in our implementation, part detection takes significantly more time than extracting CNN features. In order to increase the practicality of ELLF, part detection needs to be sped up.

Finally, instead of extracting CNN features, disjointly learning parts and then learning a classifier (an SVM), jointly learning parts with features would likely bring improvements to both part discovery and feature learning. This method would also enable us to further take advantage of data deformations, since the cost of part detection makes it impractical to train our classifier on a significant number of deformations (see Sec. IV-A). Although the advantages of jointly learning features and parts are clear, training such a non-rigid neural network efficiently (i.e. on a GPU) poses many implementation challenges – learning movable parts and pooling regions is an open problem in the design and implementation of neural networks.

V. Discussion and Future Work

In this paper we have proposed an approach for fine-grained recognition that tackles both feature learning and part discovery. Our main results are 1) that learning discriminative features in a supervised setting can be effective for fine-grained recognition, even at the small scales present in current fine-grained datasets, and 2) that one can learn parts useful for recognition without any part-level annotations. In the future we would like to combine part discovery and feature learning into a joint model and lift our part representation into 3D, which should yield more accurate correspondences.

Acknowledgments

This work was partially supported by an ONR MURI grant and the Yahoo! FREP program.

References

[1] A. B. Hillel and D. Weinshall, "Subordinate class recognition using relational object models," *NIPS*, 2007.

[2] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis, "Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance," in *ICCV*, 2011.

[3] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *CVPR*, 2012.

[4] T. Berg and P. N. Belhumeur, "POOF: Part-Based One-vs-One Features for fine-grained categorization, face verification, and attribute estimation," in *CVPR*, 2013.

[5] S. Yang, L. Bo, J. Wang, and L. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *NIPS*, 2012.

[6] B. Yao, G. Bradski, and L. Fei-Fei, "A codebook-free and annotation-free approach for fine-grained image categorization," in *CVPR*, 2012.

[7] K. Duan, D. Parikh, D. Crandall, and K. Grauman, "Discovering localized attributes for fine-grained recognition," in *CVPR*, 2012.

[8] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[9] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.

[10] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *CVPR-WFGVC*, 2011.

[11] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," Tech. Rep., 2013.

[12] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little, B. Schiele, and D. Koller, "Fine-grained categorization for 3d scene understanding," in *BMVC*, 2012.

[13] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *3dRR*, 2013.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.

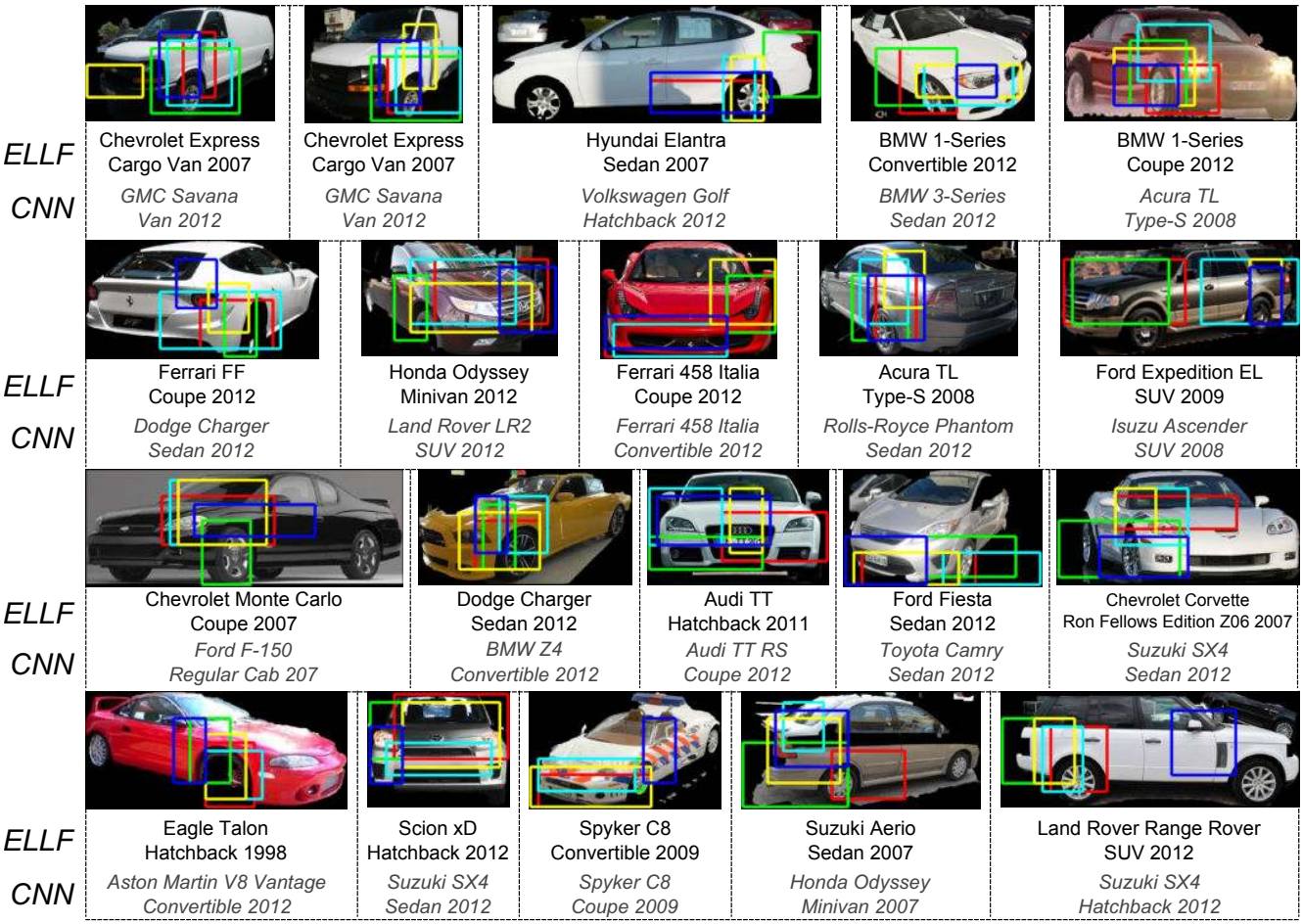

Fig. 5. Example images where ELLF was correct and a standard CNN was incorrect. On each image the five parts for our method that contributed most to a correct classification are shown. Incorrect predictions are colored gray and in italics.

Fig. 6. Example failure cases of ELLF. Incorrect predictions are colored gray and in italics. Part detection for ELLF suffers when GrabCut produces an incorrect segmentation, either by segmenting out too much of the target car or by keeping too much of the background.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[16] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *ICCV*, 2009.

[17] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *PAMI*, 2010.

[18] N. Zhang, R. Farrell, and T. Darrell, "Pose pooling kernels for sub-category recognition," in *CVPR*, 2012.

[19] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *ECCV*, 2012.

[20] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *ICCV*, 2013.

[21] Y. Chai, E. Rahtu, V. Lempitsky, L. Van Gool, and A. Zisserman, "Tricos: a tri-level class-discriminative co-segmentation method for image classification," in *ECCV*, 2012.

[22] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010.

[23] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *CVPR*, 2007.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[26] G. B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *CVPR*, 2012.

[27] G. B. Huang, M. Mattar, H. Lee, and E. G. Learned-Miller, "Learning to align from scratch," in *NIPS*, 2012.

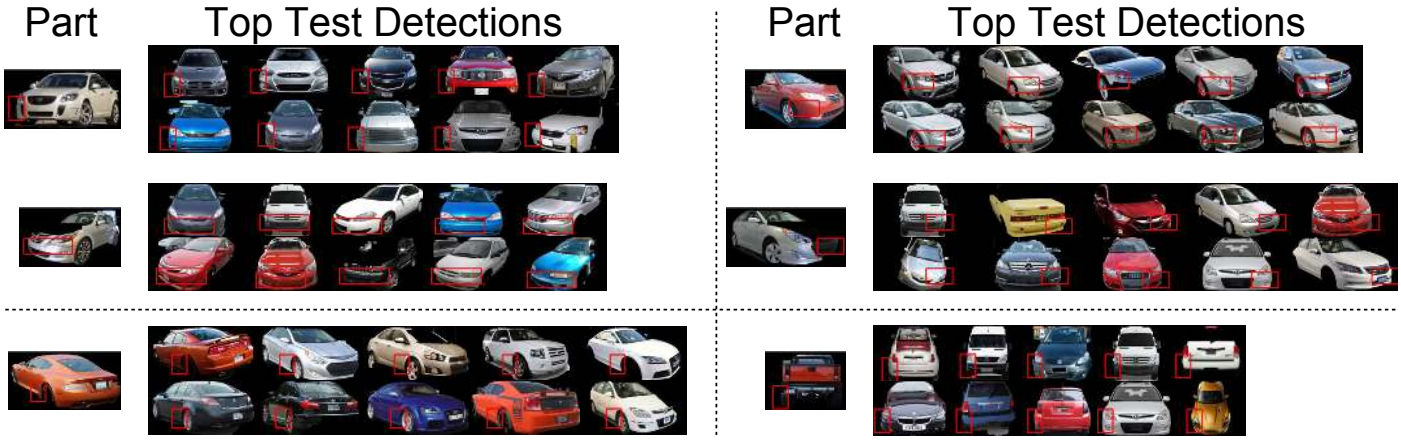

Fig. 7. A sample of parts and the ten test detections with the highest response. Each part is visualized on top of the seed image of the neighborhood which produced the trained part. The first two rows are success cases: the parts detectors fire consistently on the same parts of the car, even under the presence of some viewpoint variation. The last row are failure cases: since each part detector is based on local evidence, when different parts have the same appearance and occur in the same position in the image plane, as can occur when a car undergoes a 180-degree rotation, the part detectors misfire.

Fig. 8. The five most confusing pairs of classes for ELLF in the Car dataset, in descending order of confusion as determined by Eq. 3. We observe that these pairs of classes differ only in very small details.

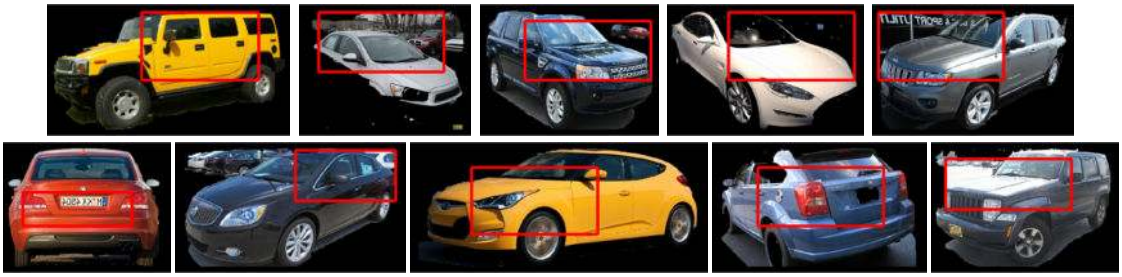

Fig. 9. The most useful parts for overall car classification. These parts tend to be large, giving information about the general type of car (SUV, sedan, etc.).

[28] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[30] M.-E. Nilsback and A. Zisserman, "Delving into the whorl of flower segmentation," in *BMVC*, 2007.

[31] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *CVPR*, 2013.

[32] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *ICCV*, 2013.

[33] C. Wah, S. Branson, P. Perona, and S. Belongie, "Multiclass recognition and part localization with humans in the loop," in *ICCV*, 2011.

[34] J. Deng, J. Krause, and L. Fei-Fei, "Fine-grained crowdsourcing for fine-grained recognition," in *CVPR*, 2013.

[35] D. Parikh and K. Grauman, "Interactively building a discriminative vocabulary of nameable attributes," in *CVPR*, 2011.

[36] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie, "Visual recognition with humans in the loop," *ECCV*, 2010.

[37] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[38] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[39] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei, "Large scale visual recognition challenge," *www.image-net.org/challenges/LSVRC/2012*.