PAPER

Efficient two-step middle-level part feature extraction for fine-grained visual categorization

Hideki NAKAYAMA[†], ***Member and*** **Tomoya TSUDA**[†], ***Nonmember***

SUMMARY Fine-grained visual categorization (FGVC) has drawn increasing attention as an emerging research field in recent years. In contrast to generic-domain visual recognition, FGVC is characterized by high intra-class and subtle inter-class variations. To distinguish conceptually and visually similar categories, highly discriminative visual features must be extracted. Moreover, FGVC has highly specialized and task-specific nature. It is not always easy to obtain a sufficiently large-scale training dataset. Therefore, the key to success in practical FGVC systems is to efficiently exploit discriminative features from a limited number of training examples. In this paper, we propose an efficient two-step dimensionality compression method to derive compact middle-level part-based features. To do this, we compare both space-first and feature-first convolution schemes and investigate their effectiveness. Our approach is based on simple linear algebra and analytic solutions, and is highly scalable compared with the current one-vs-one or one-vs-all approach, making it possible to quickly train middle-level features from a number of pairwise part regions. We experimentally show the effectiveness of our method using the standard Caltech–Birds and Stanford–Cars datasets.

key words: *Image Classification, Fine-grained Categorization, Part-based features, Dimensionality Reduction*

1. Introduction

While generic image classification techniques have been steadily progressing, fine-grained visual categorization (FGVC) [1] has remained an open problem of increasing interest. The objective of FGVC is to categorize conceptually (and thus visually) very similar classes (*e.g.*, plant and animal species) [2]–[4]. For example, Figure 1 (a) shows images of birds all belonging to different classes. It is very difficult, even for humans to correctly recognize these species. Compared with generic image recognition, FGVC is regarded as extremely difficult due to its high intra-class and low inter-class variations [2] (Figure 1), and therefore it requires new methods to capture subtle differences between categories. In general, to distinguish very similar categories, we need to extract highly informative visual features. Unlike generic visual recognition, however, sufficient training data are not always available for each specific application. This problem makes it difficult to directly apply a deep neural network [5], which is the current standard tool to extract discriminative features in generic image recognition. In some cases, deep convolutional neural networks (CNNs) substantially outperform traditional pipelines by relying on pre-trained networks on the ImageNet [6] and fine-tuning [7], [8]. However, this does not always guarantee good performance because the visual world of FGVC is often unique to each application and not always covered by generic datasets such as ImageNet.

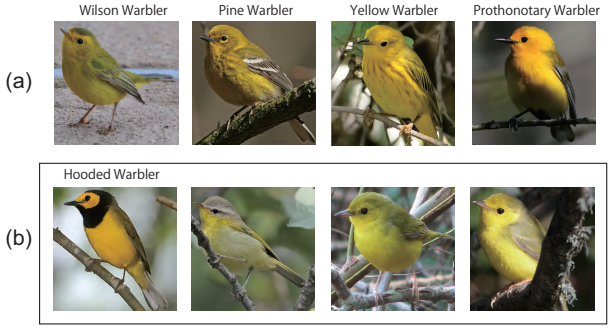


Fig. 1 Examples illustrating the difficulty of FGVC. (a) Subtle differences between categories. These birds all belong to different species. (b) Huge intra-class variations caused by the individuality of each bird such as viewpoint changes and occlusion. These birds are from the same class.

At the same time, subordinate-level categories targeted by FGVC often share some common "parts" that could provide a strong cue for recognition. For example, in Figure 1 (a), we can easily find differences if we look into the detail of the head and wing parts, even if the global shape and color of the four birds look very similar. Thus, extracting discriminative information from parts is a key concept of FGVC. Currently, part-based approaches constitute the state-of-the-art methods.

Based on this observation, we propose an efficient discriminative part-based feature extraction method. Specifically, we investigate a two-step dimensionality reduction strategy to derive discriminative middle-level part representation. To do this, we compare both space-first and feature-first convolution schemes and show their mutual effectiveness. Our approach is based on simple linear algebra and analytic solutions, and is highly scalable compared with the current one-vs-one or one-vs-all approaches to deriving middle-level features. In experiments, we evaluate our method using the standard Caltech–Birds dataset [3].

2. Related work

2.1 Various approaches to FGVC

Many approaches to FGVC have been proposed. Some early methods simply started with a well-established generic approach (*e.g.*, bag-of-visual-words [9], [10]) and achieved

Manuscript received January 1, 2011.
Manuscript revised January 1, 2011.
[†]The author is with the University of Tokyo.
DOI: 10.1587/trans.E0.??.1

Copyright © 200x The Institute of Electronics, Information and Communication Engineers

somewhat promising performance, considering the difficulty of the problem [11], [12]. However, the generic approach is intrinsically limited in that standard region features using spatial pyramids [13] are not powerful enough to capture discriminative image regions of fine-grained objects that are often highly deformable and localized. Other methods have focused on exploiting human interaction in both the training and testing phases. For example, the Visipedia system [14] uses human help in the manner of "20 questions," gradually specifying the visual characteristics of a query image that are hard for a fully automatic system to identify. In the BubbleBank system [15], crowdsourced humans point out discriminative patches within an online game. These patches are shown to substantially outperform traditional bottom-up descriptors.

As for fully automatic recognition, the accurate detection of object parts and their part-based features have substantially improved FGVC performance [16]–[18]. Roughly, this approach consists of two processes: First, parts are detected by trained part filters or object alignment methods, and then visual features are specifically extracted from each detected part. Although the part-based approach is a common strategy for generic object recognition [19], it is thought to be particularly important for FGVC. Target categories in FGVC often share a common basic structure, making part-wise comparison more reasonable. As a result, many state-of-the-art methods use a part-based approach.

While both part detection and part-based feature extraction steps are important, we specifically focus on the feature extraction.

2.2 Part-based middle-level features

Describing objects with their part-based features is a promising strategy for fine-grained recognition. However, as typical objects have a number of parts, a naive concatenation of low-level features becomes extremely high-dimensional and generally ineffective [17]. Therefore, it is important to derive middle-level discriminative part features with appropriate compression techniques. Part-based One-vs-One Features (POOF) [17] builds support vector machine (SVM) classifiers on top of low-level region features in a one-vs-one manner for random combinations of parts and classes to extract discriminative part features. Specifically, each SVM classifier projects low-level features into a scalar score of a binary classification, which is used as the middle-level feature corresponding to the specific part and class combination. This one-vs-one approach enables the mining of subtle but discriminative visual features in each part to describe the difference between very similar categories. Similarly, [20] used a one-vs-all strategy for compressing low-level part features.

3. Our approach

We assume that we have P types of parts and that the (x, y) coordinates of the center of each part are manually annotated in training data. For testing, these coordinates are estimated using appropriate part detection algorithms. An overview of our method is illustrated in Figure 2. We first extract some low-level visual features (base features) from each local grid cell in the paired-parts regions, and then apply the proposed two-step discriminative dimensionality reduction to obtain a q-dimensional middle-level feature vector for each region. Finally, all middle-level features are concatenated to represent the final feature vector for an image. Because we use two different-sized grids for base features, the dimensionality of the final image-level feature vector is $q \times {}_PC_2 \times 2 = P(P-1)q$. Finally, we train a multi-class SVM classifier using the feature vector. We further describe each step in the following sections.

3.1 Part detection and alignment

Using the part coordinates, we extract the same base features using the same methodology as that of POOF [17] to ensure a fair evaluation of the contribution of our middle-level features.

For all part combinations i, j $(i < j)$, we rotate and scale a given image so that the two parts are horizontally aligned with a horizontal spacing of 64 pixels. We then extract the 128×64 region that has the two points at its center, which we call the paired-parts region (Figure 4). In this way, we can normalize the orientation and scale of the local image regions at many different levels with respect to part combinations. While many methods just extract features from the local regions corresponding to each part, POOF improves performance by considering part combinations. However, this also results in a large number of sub regions (part-pairs) and requires an efficient dimensionality reduction scheme for processing the features of each part-pair.

3.2 Base feature extraction

From each paired-part region, we extract base features from two grids of different cell sizes; 8×8 and 16×16. We extract the following two features from each grid cell, following POOF.

1. Color histogram: We quantize the RGB-color space into 32 clusters using standard K-Means, then extract a 32-dimensional color histogram.
2. Histograms of Oriented Gradients (HOG): We use the modified HOG in [19] that constitutes a 31-dimensional feature vector for each grid cell.

For each parts-pair, extracted base features form a three-dimensional tensor, as shown in Figure 3. Note that our method is agnostic to the selection of base features and methodology for specifying local regions.

3.3 Two-step dimensionality reduction of part features

Let W_X and W_Y denote the number of grid cells along the horizontal and vertical axes, respectively, and d denote the

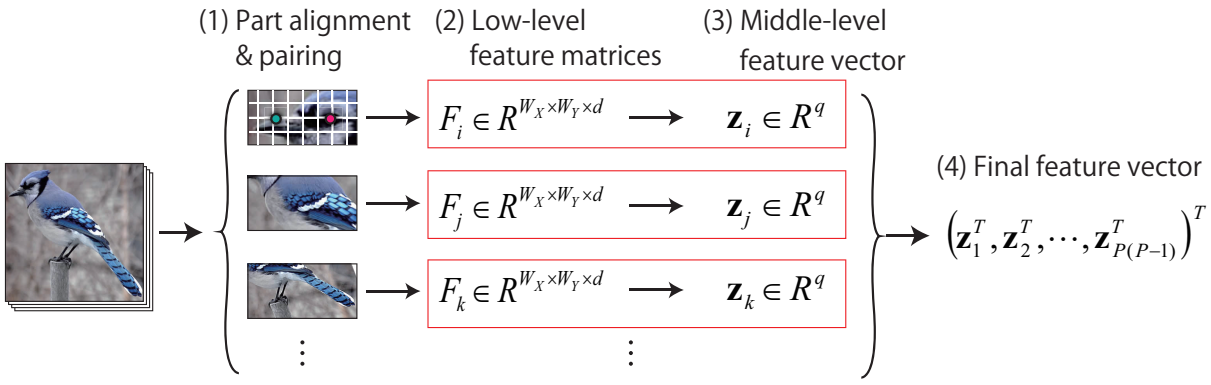


Fig. 2 Our part-based feature extraction pipeline. Red squares represent the core module contributed in this work, the detail of which is illustrated in Figure 3. (1) Extract a regularized sub-region for each part combination provided by ground-truth annotation (training) or part detector (testing). (2) Extract low-level features from grid cells in each sub-region, constituting a three-dimensional tensor. (3) Compress the tensors into a low-dimensional middle-level vector. (4) Concatenate all middle-level part features into a final image feature vector.

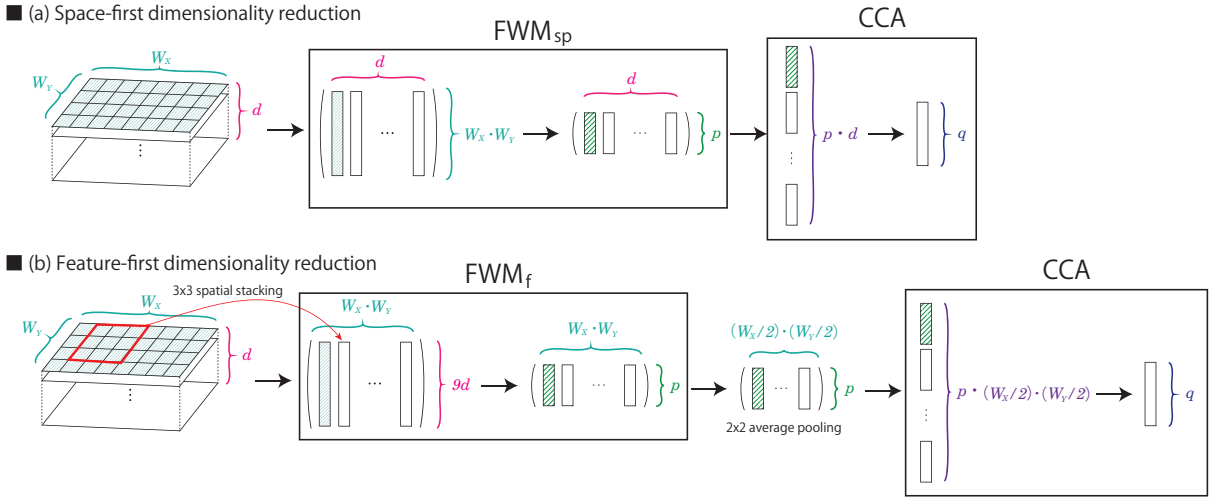


Fig. 3 Two-step discriminative dimensionality reduction of low-level part features. (a) Space-first dimensionality reduction. (b) Feature-first dimensionality reduction.

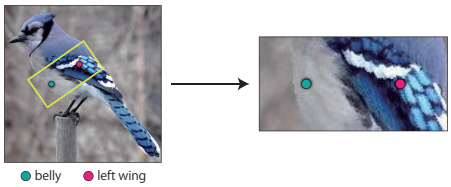


Fig. 4 Extraction of paired-parts regions. The sub-region is rotated and re-scaled so that part annotations are horizontally aligned with a specific margin.

dimensionality of base features. Figure 3 illustrates our two-step dimensionality reduction scheme. Instead of directly compressing $W_X W_Y d$-dimensional low-level descriptors, we first reduce one matrix dimension along a space or feature axis by means of a Fisher weight map (FWM). We denote these operations as FWM$_{sp}$ and FWM$_f$, respectively. After applying FWM, the reduced matrix is vectorized and canonical correlation analysis (CCA) is used to further reduce the dimension of the final feature vector using discriminative criteria. Our approach is closely related to bilinear dimensionality reduction methods such as 2D-LDA [21] and 2D-CCA [22]. A notable advantage of our method is that it is a deterministic method and no iterative optimization is required. This property enables extremely fast learning and ensures that the solution is not affected by initialization.

With regard to the feature-first dimensionality reduction (FWM$_f$), we include local convolution and pooling operations in the architecture. Namely, we stack 3×3 neighboring grid features before applying FWM. Further, we apply 2×2 spatial average pooling after FWM to reduce the number of features, making the successive CCA operation more efficient.

We let $\boldsymbol{G} \in R^{D_1 \times D_2}$ denote an input matrix for FWM. In the following derivation, we assume that the row direction is the first dimension to be compressed by FWM. Namely, we reshape an input feature tensor $\boldsymbol{F} \in R^{W_X \times W_Y \times d}$ into $\boldsymbol{G} \in R^{W_X W_Y \times d}$ for FWM$_{sp}$ and $\boldsymbol{G} \in R^{9d \times W_X W_Y}$ (with stacking) for FWM$_f$. Applying FWM, we obtain a projection matrix $\boldsymbol{W} \in R^{D_1 \times p}$ to extract the reduced feature matrix $\boldsymbol{H} \in R^{p \times D_2}$, where $\boldsymbol{H} = \boldsymbol{W}^T(\boldsymbol{G} - \bar{\boldsymbol{G}})$ ($\bar{\boldsymbol{G}}$ is the average of $\boldsymbol{G}$).

Fisher weight map (FWM)

The FWM was originally proposed by Shinohara and Otsu [23] † for computing spatial weights for individual pixels in images, and has the roots in Eigenface [25] and Fisherface [26]. While Eigenface and Fisherface simply perform principal component analysis (PCA) or Fisher linear discriminant analysis (FLDA), respectively, on image vectors, FWM is designed for a two-dimensional (matrix) representation, where each pixel has multiple feature channels.

For the formulation of FWM, let $\boldsymbol{h}$ denote a row vector of H corresponding to a single feature map $\boldsymbol{w}$, *i.e.*, $\boldsymbol{h} = \boldsymbol{w}^T(G - \bar{G})$. FWM computes the projections that maximize the Fisher's discriminant criterion of $\boldsymbol{h}$, which can be obtained as the top eigenvectors of the following generalized eigenvalue problem.

$$\Sigma_B \boldsymbol{w} = \lambda \Sigma_W \boldsymbol{w} \quad (\boldsymbol{w}^T \Sigma_W \boldsymbol{w} = 1), \tag{1}$$

where

$$\Sigma_W = \frac{1}{N} \sum_{j=1}^{C} \sum_{i=1}^{N_j} (\boldsymbol{G}_i^{(j)} - \bar{\boldsymbol{G}}^{(j)})(\boldsymbol{G}_i^{(j)} - \bar{\boldsymbol{G}}^{(j)})^T, \tag{2}$$

$$\Sigma_B = \frac{1}{N} \sum_{j=1}^{C} N_j (\bar{\boldsymbol{G}}^{(j)} - \bar{\boldsymbol{G}})(\bar{\boldsymbol{G}}^{(j)} - \bar{\boldsymbol{G}})^T. \tag{3}$$

C is the number of classes, N_j is the number of training samples in class j, $\boldsymbol{G}_i^{(j)}$ is the i-th training sample of class j, and $\bar{\boldsymbol{G}}^{(j)}$ is the class mean.

Using the top p eigenvectors with the largest eigenvalues, we compose the discriminative projection matrix $\boldsymbol{W} = \left(\boldsymbol{w}_1\ \boldsymbol{w}_2\ \cdots\ \boldsymbol{w}_p\right)$ for the first dimensionality reduction.

Canonical Correlation Analysis (CCA)

After the first compression via FWM, we vectorize the resultant matrices and apply CCA [27] to further reduce the dimensionality. We let $\boldsymbol{x} \in R^{pD_2}$ denote a vectorized (or rastered) representation of $\boldsymbol{H} \in R^{p \times D_2}$. In addition, we let $\boldsymbol{y} \in R^C$ denote its corresponding label vector, which is a one-of-K representation of the category label. Mathematically, CCA is exactly equivalent to FLDA when applied to categorization problems. One advantage of implementing CCA is that it can be easily extended to multi-label problems.

CCA finds the linear projections $s = \boldsymbol{a}^T \boldsymbol{x}$ and $t = \boldsymbol{b}^T \boldsymbol{y}$ that maximize the correlation between the projected variables s and t. We let $\Sigma = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}$ denote their covariance matrices. The CCA solution is obtained by solving the following eigenvalue problem.

$$\Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} \boldsymbol{a} = \lambda^2 \Sigma_{xx} \boldsymbol{a} \quad (\boldsymbol{a}^T \Sigma_{xx} \boldsymbol{a} = 1), \tag{4}$$

Using the top q eigenvectors for projection, we get a final part feature vector $\boldsymbol{z} = \boldsymbol{A}^T(\boldsymbol{x} - \bar{\boldsymbol{x}})$, where $\boldsymbol{A} = \left(\boldsymbol{a}_1\ \boldsymbol{a}_2\ \cdots\ \boldsymbol{a}_q\right)$.

Analysis

We compare the computational complexity of our method, PCA (CCA), POOF, and one-vs-all methods with respect to the training and extracting mid-level features. For simplicity, we only describe FWM$_{sp}$ method but the same discussion holds for FWM$_f$. The core idea of our approach is to decompose $W_X W_Y d$-dimensional eigenvalue problem of PCA (or CCA) into $W_X W_Y$-dimensional one (first step by FWM) and pd-dimensional one (second step by CCA). This approach can significantly reduce the cost from $O\left((W_X W_Y d)^3\right)$ to $O\left((W_X W_Y)^3 + (pd)^3\right)$ because we can set $p << W_X W_Y$ in general. The role of FWM is to

† Li *et al.*also proposed essentially the same method in 2005 [24].

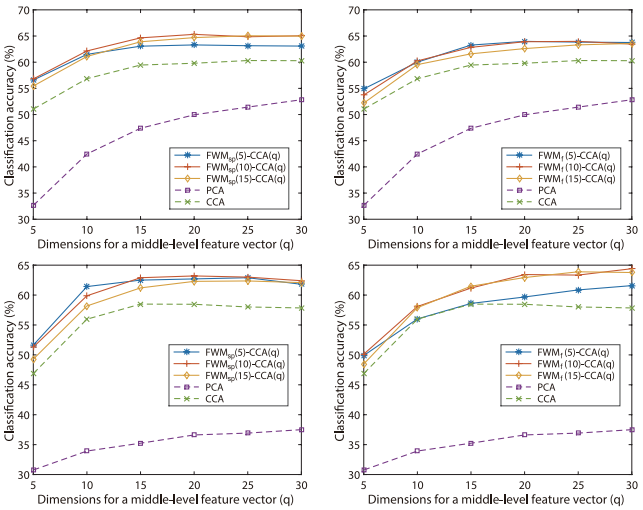


Fig. 5 Effect of two-step discriminative dimensionality reduction on low-level part features, shown together with PCA and CCA baselines. The value of q, *i.e.*, the dimensionality of a feature vector describing one part (Fig. 2), corresponds to the horizontal axes of the graphs. Top row: HOG. Bottom row: color histogram. Left column: Space-first dimensionality reduction. Right column: Feature-first dimensionality reduction.

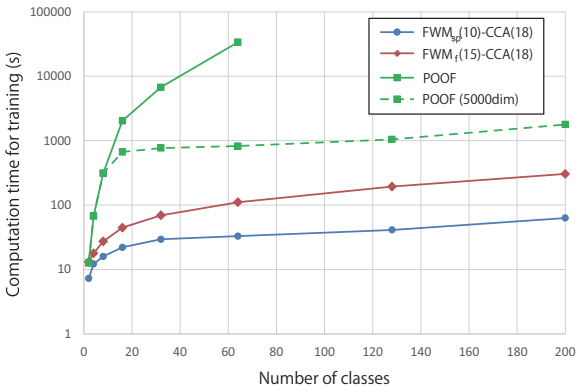


Fig. 6 Comparison of actual training time. Time for low-level feature extraction and training final SVM classifiers are not included.

roughly select pd important features so that the successive CCA is as small-dimensional as possible. Then the CCA can find final mid-level features at low cost. Also, assuming that each class has roughly the same number of training samples, our method should scale linearly to the number of classes C. Thus, our method is quite scalable compared to one-vs-all [20] and POOF [17].

As for the performance of final mid-level features, in order to get reliable performance in one-vs-one approach by POOF, we should extract all $P(P-1)C(C-1)$ combinations of parts and classes. However, this is intractable when the number of classes C gets larger. In practice, less than 1% of them are randomly sampled to make the problem feasible. More or less, the same sparseness problem is inevitable in one-vs-all approach. Unlike one-vs-one or one-vs-all, our method is based on global discriminative criterion (*i.e.*, Fisher discriminant criterion) which is suitable for extracting most discriminative features regardless of the number of classes.

4. Experiments on bird species recognition

We used the Caltech-UCSD Birds-200-2011 dataset [3] for evaluation. The dataset consists of 200 bird species, and is currently one of the most widely used benchmarks for FGVC problems. The dataset consists of 5,994 training and 5,794 test images (11,788 in total). Each image has the ground-truth annotations of a bounding box and 15 parts.

In this experiment, we tackle the "localized species categorization" benchmark, where the ground-truth bounding box may be used for both the training and testing phases [3]. Generally, classification performance is strongly affected by two components: the part estimation algorithm and part-based features. We evaluate our model under two scenarios to separately measure the individual factors. First, we use the ground-truth part locations in the test images to compare the pure performance of part-based features. The evaluation with ground-truth part locations will measure the effectiveness of the proposed feature representation under the isolation of other factors. Second, we combine our part-based features with an off-the-shelf part estimation method to evaluate the final testing performance without ground-truth part annotations.

All experiments were conducted on a workstation with two 8-core Xeon 2.60 GHz CPUs and 128 GB memory.

4.1 Dataset pre-processing

Caltech-UCSD Birds-200-2011 has annotations for the positions of 15 parts (back, beak, belly, breast, crown, forehead, left eye, left leg, left wing, nape, right eye, right leg, right wing, tail, and throat) [†]. Following the POOF methodology, and using the symmetric nature of the target (*i.e.*, bird), we reduced the complexity of the problem as follows. We horizontally reflect all images in which "right eye" is visible but "left eye" is not, and change parts "right eye/leg/wing" to "left eye/leg/wing," respectively. The part coordinates are reflected accordingly. Assuming that all targets now face leftward, we disregard the "right eye/leg/wing"" parts, which should now be occluded, and consider the remaining 12 parts in our experiments.

Although the number of original images in this dataset is not too large, remember that we consider all ${}_PC_2$ combinations of parts (${}_{12}C_2 = 66$ in our case). Each of them comprises 4,000–6,000 normalized (128×64) local images from which we extract base features of two grid sizes and train middle-level representation. In our implementation, we processed 1.3 million local images in total for part feature extraction, which is computationally intractable for unscalable methods.

[†]Note that not all parts are always visible in each image. We fill in missing parts with the average vector for that part.

Table 1 Comparison of computational complexity for extracting mid-level part features. Time for low-level feature extraction and training final SVM classifiers are not included.

	Proposed (FWM$_{sp}$)	PCA (CCA)	POOF [17]	OvA [20]
Training	$O\left(P^2\left(N((W_XW_Y)^2d+(pd)^2)+(W_XW_Y)^3+(pd)^3\right)\right)$	$O\left(P^2\left(N(W_XW_Yd)^2+(W_XW_Yd)^3\right)\right)$	$O(NP^2C^2W_XW_Yd)$	$O(NP^2CW_XW_Yd)$
Projection	$O(P^2W_XW_Yd)$	$O(P^2W_XW_Yd)$	$O(P^2C^2W_XW_Yd)$	$O(P^2CW_XW_Yd)$

Table 2 Detailed computation time for each step in training (s). Time for low-level feature extraction is not considered.

	FWM	CCA	SVM	Total	# dim	Acc. (%)
FWM$_{sp}$(10)-CCA(18)	30	32	213	275	4752	75.96
FWM$_f$(15)-CCA(18)	273	31	214	518	4752	75.27
FWM$_{sp}$(1)	57	N/A	946	1003	8316	62.11
FWM$_{sp}$(5)	190	N/A	2481	2681	41580	71.53
FWM$_{sp}$(10)	369	N/A	3776	4145	83160	72.64
FWM$_f$(1)	301	N/A	482	783	21120	61.77
FWM$_f$(5)	471	N/A	1770	2241	105600	70.78
FWM$_f$(10)	681	N/A	3255	3966	211200	72.59
CCA(18)	N/A	3084	178	3262	4752	71.90

Table 3 Classification accuracy (%) using different low-level features and compression models, together with the dimensions of final image-level representations. We set $q = 18$ (denoted as CCA(18)) except for (a5), (b5) and (c4) where we use $q = 36$ for only Opponent-SIFT.

Compression Model		No.	Color Hist.	HOG	Opp.-SIFT	# dim	Acc.(%)
FWM$_{sp}$(10)-CCA(18)		(a1)	✓			2376	63.51
		(a2)		✓		2376	65.17
		(a3)	✓	✓		4752	75.96
		(a4)	✓	✓	✓	7128	80.76
		(a5)	✓	✓	✓	9504	82.11
FWM$_f$(15)-CCA(18)		(b1)	✓			2376	64.41
		(b2)		✓		2376	64.60
		(b3)	✓	✓		4752	75.27
		(b4)	✓	✓	✓	7128	79.77
		(b5)	✓	✓	✓	9504	80.43
Combination	(a1)+(b1)	(c1)	✓			4752	65.96
(Late fusion)	(a2)+(b2)	(c2)		✓		4752	66.02
	(a3)+(b3)	(c3)	✓	✓		9504	76.39
	(a5)+(b5)	(c4)	✓	✓	✓	19008	82.15
POOF [17]		(d1)	✓	✓		5000	73.30

4.2 In-depth study using ground-truth part annotations

First, we used color histograms and HOGs as the base low-level features to demonstrate the effectiveness of our method and fairly compare it with POOF. Figures 5 plots the classification performance, reducing the compression dimensions q for HOG and color histograms. The notation FWM(p) – CCA(q) indicates that the input feature matrix is first compressed into $p \times D_2$ dimensions by FWM and successively compressed into a q-dimensional vector by CCA (see Sect. 3.3). We also plot the scores for PCA and CCA using vectorized raw features directly as the baselines. Both FWM$_{sp}$ and FWM$_f$ consistently improve performance with respect to PCA and CCA. In theory, direct CCA might seem more reasonable considering that it naturally includes our two-step linear decomposition. However, this method solves considerably high-dimensional learning problems with a limited amount of training samples, making it difficult to prevent overfitting.

Table 3 summarizes the performance of our method with various low-level features. Here, we also test Opponent-SIFT [28]. We densely sample the descriptors with exactly the same grid parameters as the others. When using multiple features, we individually fit the compression model for each feature and then concatenate all image-level features.

Comparing (a3) and (b3) with (d1) demonstrates the advantage of our methods over POOF. Using exactly the same low-level features and approximately the same dimensions of the final image feature vector, both our methods outperform POOF. This result clearly indicates the effectiveness of our mid-level feature representation. In addition, we observe that adding Opponent-SIFT features can further improve performance by a large margin. As for the comparison of (a) space-first and (b) feature-first models, we found that the former tends to perform better, although the difference is subtle. However, we observe that (c) using both of them by means of late fusion (*i.e.*, taking the average of the classifier scores) further improves classification accuracy. This result suggests that two models may exploit mutually different statistical properties of raw features.

Next, we report the computation time of our method for training (*i.e.*, building the system from 1.3M local part images). To compare our two-step method with baseline one-step approach, we summarize the detailed computation time for each step in training at Table 2, together with the dimensionality of final feature vector and classification accuracy. We omits the time for low-level feature extraction because this is common for all methods.

As we have discussed in Section 3.3, our two-step approach significantly reduces the total training time while keeping the classification accuracy. Just applying FWM ends in pD_2 features per part and does not drastically compress the dimensionality of the final feature vector for an image. Although the accuracy increases as p gets larger, training time for SVM classifiers becomes prohibitive. Also, while directly applying CCA can compress the features like two-step methods, it needs to solve $W_X W_Y d$-dimensional eigenvalue problem, resulting in long computation time for CCA. Thus, FWM and CCA in the two-step pipeline are mutually helpful to compress low-level features while keeping the cost for solving eigenvalue problems low. Our method enables fast training of discriminative middle-level representations.

Table 4 Comparison of classification accuracy when ground-truth part annotations are used for testing. Methods marked by (*) use deep learning with external training data.

Method	Acc. (%)
Ours (Color + HOG, Tab. 3 (c3))	**76.39**
Ours (Color + HOG + Opp.-SIFT, Tab. 3 (c4))	**82.15**
POOF [17]	73.30
HPM [30]	66.35
OvA(*) [20]	81.2
R-CNN(*) [7]	82.02
PN-CNN(*) [8]	85.4

We also evaluated the scalability in terms of the number of classes. Figure 6 summarizes the result. As we have described in Table 1, our method should scale linearly to the number of classes, considering that each class has roughly the same number of samples. On the other hand, POOF quickly becomes intractable as the number of classes grows. Although we can cap the number of features by random sampling (dashed line shows when capped at 5,000) as in [17], this will lead to significant loss of performance because the sampled features may become extremely sparse when the number of classes is large.

Finally, we summarize the results of our method and previous work in Table 4. Notation (*) indicates that the method requires external training data (ImageNet) to train (or pre-train) deep CNNs. Starting from off-the-shelf low-level descriptors, our method achieves promising performance well comparable with those using fine-tuned CNNs.

4.3 Classification performance including part estimation

We next evaluate the final performance of our method, including the estimation of part locations. Because part localization is not the main interest of this study, but code for the part detection employed in POOF is not publicly available, we use the very simple but efficient method of Nonparametric Part Transfer (NPT) [29]. NPT simply transfers the part locations of the nearest neighbors of the input query in terms of the global HOG descriptor, as follows.

1. Augment the training dataset by including horizontally reflected images [†]. Part locations (and types) are accordingly reflected.
2. For a test image, retrieve the k nearest neighbors in terms of HOG-space distance and impose each of their part locations.

[†]Note that augmenting the dataset by adding horizontally reflected images is a common practice, *e.g.*, in [31], [32].

3. Using the transferred part annotations, extract part-based features and conduct classification.
4. Final classification is decided by means of the late fusion of classification scores corresponding to k neighbors.

Table 5 summarizes the results when ground-truth part locations are unknown. In addition to our methods and previous work, we also report the scores using the Fisher vector [10] with the Opponent-SIFT descriptor. To implement the Fisher vector, we densely sampled the descriptors from 24×24 patches at every three pixels on a grid, and encoded them using a Gaussian mixture model with 64 components. We extracted features from four regions: the entire image and three horizontal bins. Model (a1) in Table 5 corresponds to the standard implementation. In addition, we also extracted features and trained a classifier from the segmented images using GrabCut [33], combining it with (a1) by means of late fusion, which we denote by (a2). These baseline models were trained on the dataset augmented by reflection.

Our method with NPT part estimation (b2) achieves 47.31%. This is relatively poor compared with the original score of 56.78% reported in POOF. The difference in performance may be attributed to the choice of part estimation method. However, late fusion with the global Fisher vector model (a2) greatly improves the performance to 64.08% (b3), which is state-of-the-art among those methods not using CNNs (and external training data). Using a more powerful part localization method could further improve performance.

5. Experiments on car type recognition

We also evaluated our method on the Stanford Cars-196 dataset [35] covering 196 fine-grained car types. The dataset contains 8,144 training and 8,041 test images specified by the authors.

Basically, our method assumes ground-truth annotations of part locations for training. However, this dataset has only bounding box information and does not have part annotations. Therefore, we use the deformable part model (DPM) [19] to automatically estimate parts both for training and testing images. We use the car model pre-trained on the VOC2007 dataset provided by the authors[†]. Unlike in the experiment on the Birds-dataset, we do not perform any preprocessing like section 4.1. Also, instead of NPT, we directly use the part locations estimated by DPM for the testing phase.

We summarize the classification accuracies of our methods and previous ones in Table 6. As the global feature baseline, we found that the SIFT-based Fisher vector achieves the best (71.0%). Adding our mid-level part features by late fusion can improve 1.6%, despite that both of them use exactly the same local descriptors (*i.e.*, dense SIFT). Moreover, adding more descriptors for computing part features can further improve the performance and obtained 73.8% accuracy. This is close to ELLF [36], the current best result using no external training data. Considering that ELLF depends on a well-tuned CNN with significant data augmentation, our result is quite satisfactory. Thus, we conclude that our method is also effective for car type recognition.

[†]http://www.cs.berkeley.edu/ rbg/latent/

6. Conclusion

In this study, we tackled the problem of fine-grained visual categorization. The key to success in the part-based approach is to derive discriminative and compact middle-level representation of part features. The core contribution of our work is the efficient learning of middle-level features using a two-step dimensionality reduction scheme. In contrast to one-vs-one or one-vs-all alternatives, our method is based on analytic and deterministic solutions using global cost functions, which are scalable in terms of the number of classes. Moreover, the two-step decomposition reduces the dimension of eigenvalue problems and thus enables extremely fast training. Although this family of methods is widely studied in the face-recognition domain, we found these properties are quite beneficial for FGVC problems where we need to handle a number of classes and part regions.

In experiments using the standard Caltech-Birds and Stanford-Cars dataset, we confirmed that our middle-level features substantially outperform the POOF baseline, despite their simplicity. Moreover, used together with off-the-shelf part detection methods, our approach achieved similar or better results than previous state-of-the-art methods that do not use external data for deep learning.

Acknowledgments

This work was supported by JST CREST, JSPS KAKENHI Grant Number 26730085 and the Kayamori Foundation of Informational Science Advancement.

References

[1] I. Biederman, S. Subramaniam, and M. Bar, "Subordinate-level object classification reexamined," Psychological Research, vol.62, pp.131–153, 1999.
[2] J. Deng, A. Berg, K. Li, and L. Fei-Fei, "What does classifying more than 10,000 image categories tell us?," Proc. ECCV, pp.71–84, 2010.
[3] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," tech. rep., California Institute of Technology, 2011.
[4] M.E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," Proc. Indian Conference on Computer Vision, Graphics & Image Processing, pp.722–729, Ieee, Dec. 2008.
[5] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional neural networks," Proc. NIPS, 2012.
[6] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," Proc. CVPR, pp.2–9, 2009.

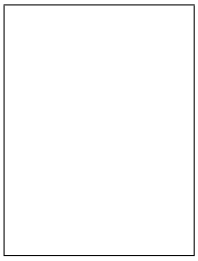
Table 5 Comparison of classification accuracy including automatic part detection. Methods marked by (*) use deep learning with external training data.

Method	No.	Part detection	Base part/region features	Obj.-level features	Acc. (%)
Baseline	(a1)	-	-	Opp.-SIFT FV	51.4
	(a2)	-	-	Opp.-SIFT FV (Raw + GrabCut)	56.2
Ours	(b1)	NPT	Color hist + HOG	-	42.42
	(b2)	NPT	Color hist + HOG + Opp.-SIFT	-	47.31
	(b3)	NPT	Color hist + HOG + Opp.-SIFT	Opp.-SIFT FV (Raw + GrabCut)	**64.08**
POOF [17]	(c1)	Exemplars	Color hist + HOG	-	56.78
NPT [29]	(c2)	NPT	Color name BoW + Opp.-SIFT BoW		57.84
DPD [18]	(c3)	DPM	Kernel descriptor BoW		50.98
HPM [30]	(c4)	DPM + GrabCut	Opp.-SIFT LLC		59.86
Symbiotic [32]	(c5)	DPM + GrabCut	SIFT FV + Color hist LLC		59.40
Alignment [31]	(c6)	Unsup. alignment	Opp.-SIFT FV		**62.70**
DeCAF(*) [34]	(d1)	DPM	CNN		64.96
OvA(*) [20]	(d2)	DPM + GrabCut	CNN + SIFT FV + Color LLC	-	67.6
R-CNN(*) [7]	(d3)	Region proposal	CNN		**76.37**
PN-CNN(*) [8]	(d4)	Pose normalization	CNN		75.7

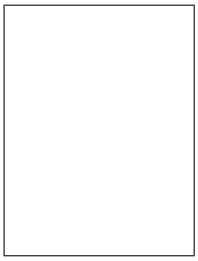
Table 6 Comparison of classification accuracy on the Stanford Cars-196 dataset. Ground-truth bounding box annotations are used both for training and testing.

Method	Acc. (%)
FV (SIFT)	71.0
Ours (SIFT) + FV (SIFT)	**72.6**
Ours (Color, HOG, Opp.-SIFT, SIFT) + FV (SIFT)	**73.8**
BB [15]	63.6
BB-3D-G [35]	67.6
LLC (SIFT) [37]	69.5
CNN [36]	70.5
ELLF [36]	73.9

[7] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for Fine-grained Category Detection," Proc. ECCV, 2014.
[8] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird Species Categorization Using Pose Normalized Deep Convolutional Nets," Proc. BMVC, 2014.
[9] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," Proc. ECCV Workshop on Statistical Learning in Computer Vision, 2004.
[10] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," Proc. ECCV, 2010.
[11] J. Lin and T.G. Dietterich, "Is Fine Grained Classification Different ?," CVPR workshop on FGVC, 2013.
[12] P.h. Gosselin, N. Murray, H. Jégou, and F. Perronnin, "Revisiting the Fisher vector for fine-grained classification," Pattern Recognition Letters, vol.11, no.49, pp.92–98, 2014.
[13] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," Proc. IEEE CVPR, 2006.
[14] S. Branson, C. Wah, and F. Schroff, "Visual recognition with humans in the loop," Proc. ECCV, 2010.
[15] J. Deng, J. Krause, and L. Fei-Fei, "Fine-Grained Crowdsourcing for Fine-Grained Recognition," Proc. IEEE CVPR, 2013.
[16] N. Zhang, R. Farrell, and T. Darrell, "Pose Pooling Kernels for Subcategory Recognition," Proc. IEEE CVPR, 2012.
[17] T. Berg and P.N. Belhumeur, "POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.955–962, 2013.
[18] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable Part Descriptors for Fine-Grained Recognition and Attribute Prediction," Proc. IEEE ICCV, 2013.
[19] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminative Trained Part Based Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.32, pp.1627–1645, 2010.
[20] X. Zhang, H. Xiong, W. Zhou, and Q. Tian, "Fused One-vs-All Mid-Level Features for Fine-Grained Visual Categorization," Proc. ACM Multimedia, 2014.
[21] J. Ye, R. Janardan, and Q. Li, "Two-Dimensional Linear Discriminant Analysis," Proc. NIPS, 2004.
[22] S.H. Lee and S. Choi, "Two-Dimensional Canonical Correlation Analysis," IEEE Signal Processing Letters, vol.14, no.10, pp.735–738, 2007.
[23] Y. Shinohara and N. Otsu, "Facial expression recognition using Fisher weight maps," IEEE FG, 2004.
[24] M. Li and B. Yuan, "2D-LDA: A statistical linear discriminant analysis for image matrix," Pattern Recognition Letters, vol.26, no.5, pp.527–532, 2005.
[25] M. Turk and A. Pentland, "Face recognition using eigenfaces," Proc. IEEE CVPR, 1991.
[26] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," IEEE Trans. PAMI, vol.19, no.7, pp.711–720, 1997.
[27] H. Hotelling, "Relations between two sets of variants," Biometrika, vol.28, pp.321–377, 1936.
[28] K.E.a. van de Sande, T. Gevers, and C.G.M. Snoek, "Evaluating color descriptors for object and scene recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.32, no.9, pp.1582–96, Sept. 2010.
[29] G. Christoph, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric Part Transfer for Fine-grained Recognition," Proc. IEEE CVPR, 2014.
[30] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, "Hierarchical part matching for fine-grained visual categorization," IEEE ICCV, pp.1641–1648, 2013.
[31] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, and T. Tuytelaars, "Fine-Grained Categorization by Alignments," Proc. IEEE ICCV, 2013.
[32] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic Segmentation and Part Localization for Fine-Grained Categorization," Proc. IEEE ICCV, 2013.
[33] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut" - Interactive foreground extraction using iterated graph cuts," ACM Transactions on Graphics (SIGGRAPH), 2004.
[34] 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," Proc. ICML, 2014.
[35] J. Krause, M. Stark, J. Deng, and L. Fei-fei, "3D Object Representations for Fine-Grained Categorization," IEEE Workshop on 3D Representation and Recognition, 2013.
[36] J. Krause, T. Gebru, J. Deng, L.J. Li, and F.F. Li, "Learning Features and Parts for Fine-Grained Recognition," Proc. ICPR, 2014.
[37] J. Wang, J. Yang, K. Yu, F. Lv, T.S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," Proc. IEEE CVPR, pp.3360–3367, Ieee, June 2010.



Hideki Nakayama received the M.S. and Ph.D degrees in information science from the University of Tokyo in 2008 and 2011, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science (DC1) from 2008 to 2011. He is currently a full-time senior assistant professor at the Graduate School of Information Science and Technology, The University of Tokyo. His research interests include generic object and image recognition, multimedia analysis, natural language processing and deep learning.



Tomoya Tsuda received his M.S. degree in information science from the University of Tokyo in 2015. His research interests include image classification and machine learning. He is currently at Shimadzu Corporation.