Blocks that Shout:
Distinctive Parts for Scene Classification

Mayank Juneja[1]   Andrea Vedaldi[2]   C. V. Jawahar[1]   Andrew Zisserman[2]

[1] Center for Visual Information Technology, International Institute of Information Technology, Hyderabad, India

[2] Department of Engineering Science, University of Oxford, United Kingdom

{mayank.juneja@research.,jawahar@}iiit.ac.in   {vedaldi,az}@robots.ox.ac.uk

Abstract

The automatic discovery of distinctive parts for an object or scene class is challenging since it requires simultaneously to learn the part appearance and also to identify the part occurrences in images. In this paper, we propose a simple, efficient, and effective method to do so. We address this problem by learning parts incrementally, starting from a single part occurrence with an Exemplar SVM. In this manner, additional part instances are discovered and aligned reliably before being considered as training examples. We also propose entropy-rank curves as a means of evaluating the distinctiveness of parts shareable between categories and use them to select useful parts out of a set of candidates.

We apply the new representation to the task of scene categorisation on the MIT Scene 67 benchmark. We show that our method can learn parts which are significantly more informative and for a fraction of the cost, compared to previous part-learning methods such as Singh et al. [28]. We also show that a well constructed bag of words or Fisher vector model can substantially outperform the previous state-of-the-art classification performance on this data.

1. Introduction

The notion of *part* has been of central importance in object recognition since the introduction of pictorial structures [11], constellation models [35], object fragments [2, 29], right up to recent state-of-the-art methods such as Deformable Part Models (DPMs) [9]. Yet, the *automatic discovery* of good parts is still a difficult problem. In DPM, for example, part occurrences are initially assumed to be in a fixed location relative to the ground truth object bounding boxes, and then are refined as latent variables during learning [9]. This procedure can be satisfactory in datasets such as PASCAL VOC [8] where bounding boxes usually induce a good alignment of the corresponding objects. However,

when the alignment is not satisfactory, as for the case of highly-deformable objects such as cats and dogs [20], this approach does not work well and the performance of the resulting detector is severely hampered.

In this paper, a simple, efficient, and effective method for discovering parts automatically and with very little supervision is proposed. Its power is demonstrated in the context of scene recognition where, unlike in object recognition, object bounding boxes are not available, making part alignment very challenging. In particular, the method is tested on the MIT Scene 67 dataset, the standard benchmark for scene classification. Fig. 1 shows examples of the learned parts detected on the test set.

To achieve these results two key issues must be addressed. The first is to find and align part instances in the training data while a model of the part is not yet available. This difficulty is bypassed by learning the model from a *single exemplar* of a candidate part. This approach is motivated by [17], that showed that a single example is often sufficient to train a reasonable, if a little restrictive, detector. This initial model is then refined by alternating mining for additional part instances and retraining. While this procedure requires training a sequence of detectors, the LDA technique of [13] is used to avoid mining for hard negative examples, eliminating the main bottleneck in detector learning [9, 32], and enabling a very efficient part-learning algorithm.

The second issue is to select *distinctive parts* among the ones that are generated by the part mining process. The latter produces in fact a candidate part for each of a large set of initial seeds. Among these, the most informative ones are identified based on the novel notion of *entropy-rank*. This criterion selects parts that are informative for a small proportion of classes. Differently to other measures such as average precision, the resulting parts can then be *shared* by more than one object category. This is particularly important because parts should be regarded as mid-level primitives that do not necessarily have to respond to a *single* object class.
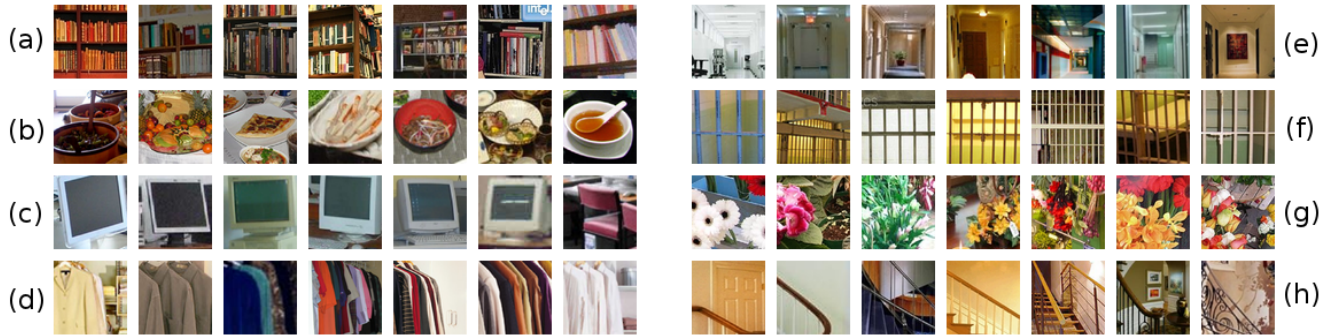
Figure 1. Example of occurrences of distinctive parts learned by our method from weakly supervised image data. These part occurrences are detected on the test data. (a) bookstore, (b) buffet, (c) computerroom, (d) closet, (e) corridor, (f) prisoncell, (g) florist, (h) staircase.

The result of our procedure is the automatic discovery of distinctive part detectors. We call them "blocks that shout" due to their informative nature and due to the fact that, in practice, they are implemented as HOG [7] block filters (Fig. 2).

Related work. Parts have often been sought as intermediate representations that can complement or substitute lower level alternatives like SIFT [16]. In models such as DPMs, parts are devoid of a specific semantic content and are used to represent deformations of a two dimensional template. Often, however, parts do have a semantic connotation. For example, in Poselets [5] object parts correspond to recognizable clusters in appearance and configuration, in Li *et al.* [15] scene parts correspond to object categories, and in Raptis *et al.* [25] action parts capture spatio-temporal components of human activities.

The learning of parts is usually integrated into the learning of a complete object or scene model [3, 9]. Only a few papers deal primarily with the problem of learning parts. A possibility, used for example by Poselets [5], is to use some spatial annotation of the training images for weakly supervised learning. Singh *et al.* [28] explores learning parts in both an unsupervised and weakly supervised manner, where the weakly supervised case (as here) only uses the class label of the image. Their weakly supervised procedure is applied to the MIT Scene 67 dataset, obtaining state-of-the-art scene classification performance. As will be seen though, our part-learning method is: (i) simpler, (ii) more efficient, and (iii) able to learn parts that are significantly better at scene classification.

The problem of scene classification has been approached in other papers from a variety of different angles. For example, Sadeghi and Tappen [26] address this problem with a representation based on discriminative scene regions. Parizi *et al.* [19] propose a reconfigurable version of a spatial bag of visual words (BoW) model that associates different BoW descriptors to different image segments, corresponding to different types of "stuff". Wu and Rehg [36] propose a novel holistic image descriptor for scene classification. The standard DPM model is applied to the task of scene categorization by Pandey and Lazebnik [18], but the problem of part initialization and learning is not addressed, and the quality of the parts that can be obtained in this manner remains unclear. Li *et al.* [15] apply their object bank, and Zhu *et al.* [37] explore the problem of jointly modeling the interaction of objects and scene topics in an upstream model, where topics are semantically meaningful and observed. Quattoni and Torralba [24] study the problem of modeling scenes layout by a number of prototypes capturing characteristic arrangements of scene components. Interestingly, one of the results of this paper is that *a well designed BoW or Fisher Vector [21] model trained on a single feature* can beat all these approaches.

2. Blocks that shout: learning distinctive parts

In characterizing images of particular scene classes, *e.g.* a computer room, library, book store, auditorium, theatre, etc., it is not hard to think of distinctive parts: chairs, lamps, doors, windows, screens, etc., readily come to mind. In practice, however, a distinctive part is useful only if it can be detected automatically, preferably by an efficient and simple algorithm. Moreover, distinctive parts may include other structures that have a weaker or more abstract semantic, such as the corners of a room or a corridor, particular shapes (rounded, square), and so on. Designing a good vocabulary of parts is therefore best left to *learning*.

Learning a distinctive part means identifying a localized detectable entity that is informative for the task at hand (in our example discriminating different scene types). This is very challenging because (i) one does not know if a part occurs in any given training image or not, and (ii) when the part occurs, one does not know its location. While methods such as multiple instance learning have often been proposed to try to identify parts automatically, in practice they require careful initialization to work well.

Our strategy for part-learning combines three ideas: seeding, expansion, and selection. In *seeding* (Sect. 2.1) a set of candidate part instances (seeds) is generated by sam-
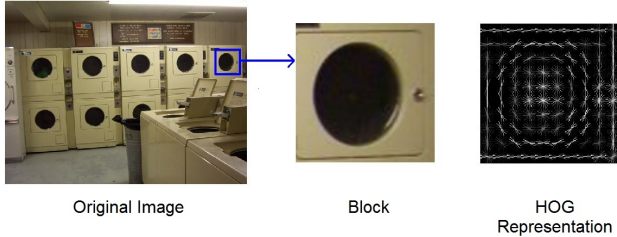
Figure 2. An example block learnt automatically for the laundro-mat class. This block is a characteristic part for this class.

pling square windows (blocks) in the training data guided by segmentation cues. All such blocks are treated initially as potentially different parts. In *expansion*, (Sect. 2.2) each block is used as a seed to build a model for a part while gradually searching for more and more part occurrences in the training data. This paced expansion addresses the issue of detecting and localizing part exemplars. Finally, *selection* (Sect. 2.3) finds the most distinctive parts in the pool of candidate parts generated by seeding and expansion by looking at their predictive power in terms of entropy-rank. The procedure is weakly supervised in that positives are only sought in the seeding and expansion stages within images of a single class.

Once these distinctive parts are obtained, they can be used for a variety of tasks. In our experiments they are encoded in a similar manner to BoW: an image descriptor is computed from the maximum response of each part pooled over spatial regions of the image (Sect. 3.2). Experiments show that this representation is in fact *complementary* to BoW (Sect. 4).

2.1. Seeding: proposing an initial set of parts

Initially, no part model is available and, without further information, any sub-window in any training image is equally likely to contain a distinctive part. In principle, one could simply try to learn a part model by starting from *all possible image sub-windows* and identify good parts *a-posteriori*, during the selection stage (Sect. 2.3). Unfortunately, most of these parts will in fact not be distinctive (*e.g.* a uniform wall) so this approach is highly inefficient.

We use instead low-level image cues to identify a subset of image locations that are more likely to be centered around distinctive parts. Extending the notion of *objectness* [1], we say that such promising locations have high *partness*. In order to predict partness, we use image oversegmentations, extending the idea of [30] from objects to parts.

In detail, each training image is first segmented into superpixels by using the method of [10]. This is repeated four times, by rescaling the image with scaling factors $2^{-\frac{i}{3}}, i = 0, 1, 2, 3$. Superpixels of intermediate sizes, defined as the ones whose area is in the range 500 to 1,500 pixels, are retained. These threshold are chosen assuming

that the average area of an unscaled image is 0.5 Mpixels. Superpixels which contain very little image variation (as measure by the average norm of the intensity gradient) are also discarded.

Part models are constructed on top of HOG features [7]. At each of the fours scales, HOG decomposes the image into cells of 8×8 pixels. Each part is described by a block of 8×8 HOG cells, and hence occupies an area of 64×64 pixels. A part seed is initialized for each superpixel by centering the 64×64 pixel block at the center of mass of the superpixel. Compared to sampling blocks uniformly on a grid, this procedure was found empirically to yield a much higher proportion of seeds that successfully generate good discriminative parts.

Figure 3 shows an example of the superpixels computed from a training image, and the seed blocks obtained using this procedure.

2.2. Expansion: learning part detectors

Learning a part detector requires a set of part exemplars, and these need to be identified in the training data. A possible approach is to sample at random a set of part occurrences, but this is extremely unlikely to hit multiple occurrences of the same part. In practice, part initialization can be obtained by means of some heuristic, such as clustering patches, or taking parts at a fixed location assuming that images are at least partially aligned. However, the detector of a part is, by definition, the most general and reliable tool for the identification of that part occurrences.

There is a special case in which a part detector can be learned without worrying about exemplar alignment: a training set consisting exactly of one part instance. It may seem unlikely that a good model could be learned from a single part example, but Exemplar SVMs [17] suggest that this may be in fact be the case. While the model learned from a single occurrence cannot be expected to generalise much, it is still sufficient to identify reliably at least a few other occurrences of the part, inducing a gradual expansion process in which more and more part occurrences are discovered and more variability is learned.

In practice, at each round of learning the current part model is used to rank blocks from images of the selected class and the highest scoring blocks are considered as further part occurrences. This procedure is repeated a set number of times (ten in the experiments), adding a small number of new part exemplars (ten) to the training set each time. All the part models obtained in this manner, including the intermediate ones, are retained and filtered by distinctiveness and uniqueness in Sect. 2.3. Figure 4 shows an example seed part on the left, and the additional part occurrences that are added to the training set during successive iterations of expansion.

Note that this expansion process uses a *discriminative*

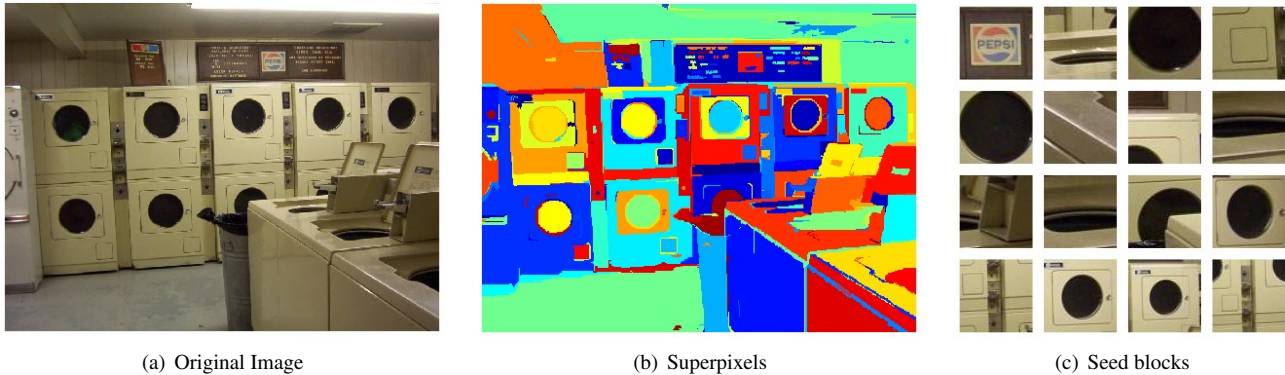(a) Original Image   (b) Superpixels   (c) Seed blocks

Figure 3. **Selecting seed blocks.** The super-pixels (b) suggest characteristic regions of the image, and blocks are formed for these. Blocks with low spatial gradient are discarded.

model of the part. This is particularly important because in image descriptors such as HOG most of the feature components correspond to irrelevant or instance specific details. Discriminative learning can extract the distinctive information (*e.g.* shape), while generative modeling (*e.g.* k-means clustering) has difficulty in doing so and constructing "semantic" clusters.

LDA acceleration. The downside of this mining process is that the part detector must be learned multiple times. Using a standard procedure that involves hard negative mining for each trained detector [9, 32] would then be very costly. We use instead the LDA technique of [13], which can be seen as learning *once* a soft but universal model of *negative* patches (a similar method is described in [12]). In practice, the parameter vector w of a part classifier is learned simply as $w = \Sigma^{-1}(\bar{x} - \mu_0)$ where $\bar{x}$ is the mean of the HOG features of the positive part samples, μ_0 is the mean of the HOG blocks in the dataset, and Σ the corresponding covariance matrix. HOG blocks are searched at all locations at the same four scales of Sect. 2.1.

2.3. Selection: identifying distinctive parts

Our notion of a discriminative block is that it should occur in many of the images of the class from which it is learnt, but not in many images from other classes. However, it is not reasonable to assume that parts (represented by blocks) are so discriminative that they only occur in the class from which they are learnt. For example, the door of a washing machine will occur in the laundromat class, but can also occur in the kitchen or garage class. Similarly, a gothic arch can appear in both the church and cloister class. However, one would not expect these parts to appear in many other of the indoor classes. In contrast, a featureless wall could occur in almost any of the classes.

In selecting the block classifiers we design a novel measure to capture this notion. The block classifiers were learnt on training images for a particular class, and they are tested as detectors on validation images of all classes. Blocks

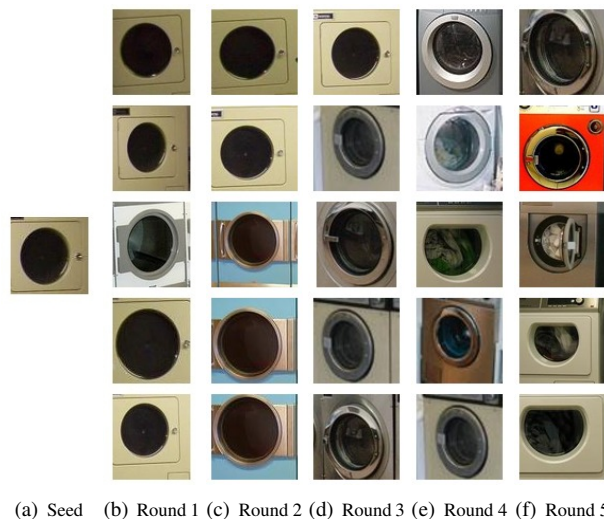(a) Seed   (b) Round 1 (c) Round 2 (d) Round 3 (e) Round 4 (f) Round 5

Figure 4. **Mining of part instances.** The seed (initial) block is on the left. On the right the additional example blocks added to the positive training set for retraining the part detector are shown in the order that they are added. Note that mining uses blocks selected from a certain scene category, but no other supervision is used.

learned from a class are not required to be detected only from images of that class; instead, the milder constraint that the distribution of classes in which the block is detected should have low entropy is imposed. In this manner, distinctive but shareable mid-level parts can be selected. For the laundromat example above, we would expect the washing machine door to be detected in only a handful of the classes, so the entropy would be low. In contrast the block for a wall would be detected across many classes, so its distribution would be nearer uniform across classes, and hence the entropy higher.

To operationalize this requirement, each block is evaluated in a sliding-window manner on each validation image. Then, five block occurrences are extracted from each image by max-pooling in five image regions, corresponding to the spatial subdivisions used in the encoding of Sect. 3. Each
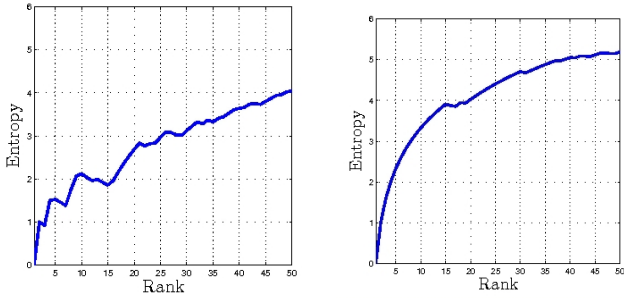
(a) Discriminative detector   (b) Non-Discriminative detector

Figure 5. **Entropy-Rank (ER Curves).** Entropy values reach a uniform value as more and more images are ranked, with images from different classes coming in. Classifier (a) has low entropy at top ranks, which shows that it is picking the blocks from a few classes. On the other hand, Classifier (b) has more uniform entropy, which shows that it is picking the blocks from many classes, making the classifier less discriminative. For each class, classifiers with low AUC are selected.

block occurrence (z_i, y_i) detected in this manner receives a detection score z and a class label y equal to the label of the image. The blocks are sorted on their score z, and the top r ranking blocks selected. Then the entropy $H(Y|r)$ is computed:

$$H(Y|r) = -\sum_{y=1}^{N} p(y|r) \log_2 p(y|r), \qquad (1)$$

where N is the number of image classes and $p(y|r)$ is the fraction of the top r blocks (z_i, y_i) that have label $y_i = y$. We introduce *Entropy-Rank Curves (ER curves)* to measure the entropy of a block classifier at different ranks. An ER curve is similar to a Precision-Recall Curve (PR curve), with rank on the x-axis and entropy values on the y-axis.

Figure 5 shows the ER curves of a discriminative and a non-discriminative block detector, respectively. Note, entropy for all part classifiers converges to a constant value (which depends on the class prior) as the rank increases. Analogously to Average Precision, we then take the Area Under Curve (AUC) for the ER graph as an overall measure of performance of a detector. The top scoring detectors based on this measure are then retained.

The final step is to remove redundant part detectors. In fact, there is no guarantee that the part mining procedure will not return the same or similar parts multiple times. The redundancy between a pair of detectors w' and w'' is measured by their cosine similarity $\langle w'/\|w'\|, w''/\|w''\| \rangle$. For each class, n detectors are selected sequentially by increasing ER scores, skipping detectors that have cosine similarity larger than 0.5 with any of the detectors already selected.

3. Image representations and learning

The part detectors developed in Sect. 2 are used to construct "bag of parts" image-level descriptors. These can be used in combination with BoW or Fisher Vector descriptors (by stacking the corresponding vectors after normalization). The details of these representations are given in Sect. 3.1– Sect. 3.2 respectively, and learning and classification is described in Sect. 3.3.

3.1. Bag of parts

In order to compute an image-level descriptor from the parts learned in Sect. 2, all the corresponding classifiers are evaluated densely at every image location at multiple scales. Part scores are then summarized in an image feature vector by using max-pooling, by retaining the maximum response score of a part in a region. The pooling is done in a spatial-pyramid fashion [14] (1×1, 2×2 grids), and encodings of each spatial region are stacked together to form the final image representation of the image (the "bag of parts"). Note that the method of selecting the parts (Sect. 2.3) and here the encoding into an image feature vector both use max-pooling over the spatial-pyramid.

3.2. Bag of visual words and Fisher vectors

We investigate a number of feature encodings, as described in [6]: (i) hard assignment (*i.e.* vector quantization) BoW; (ii) kernel-codebook encoding BoW [22, 31]; (iii) Locality-constrained Linear Coding (LLC) BoW [34]; and (iv) Improved Fisher Vectors (IFV) [21]. Each encoding uses the parameters given in [6] unless otherwise specified.

Dense visual words. Dense RootSIFT descriptors [4, 16] are extracted from the image with a spatial stride of three to five pixels and at six scales, defined by rescaling with factors $2^{-\frac{i}{2}}, i = 0, 1, \ldots, 5$. Low-contrast SIFT descriptors (identified as the ones for which the average intensity gradient magnitude is below a threshold) are mapped to null vectors. The RootSIFT descriptors are then mapped to visual words. For the IFV encoding, the visual word dictionary is obtained by training a Gaussian Mixture Model (diagonal covariance) with 256 centers; for the other encodings, the dictionary is learned by using k-means and setting k to 2,000.

Spatial encoding. Weak geometric information is retained in the descriptors by using spatial histogramming [14]. For the IFV encoding, the image is divided into 1×1, and 2×2 grids, obtaining a total of 5 spatial pooling regions; for the other encodings, the image is divided into 1×1, and 2×2, and 4×4 grids, obtaining a total of 21 spatial pooling regions. The descriptors for each region are individually normalized, and then stacked together to give the final image descriptor. For the IFV encoding, the image descriptor is

204,800-dimensional, and for other encodings, the image descriptor is 42,000-dimensional.

3.3. Learning and classification

Learning uses the PEGASOS SVM [27] algorithm, a linear SVM solver. In order to use non-linear additive kernels instead of the linear one, the χ^2 explicit feature map of [33] is used (the bag of parts and bag of words histograms are l^1 normalized). Using the feature map increases the dimension of the input feature vector by 3 times. For the IFV encoding, we use square-root (Hellinger) kernel. The parameter C of the SVM (regularization-loss trade-off) is determined by 4-fold cross validation. For multi-class image classification problems, 1-vs-rest classifiers are learned. In this case, it was found beneficial to calibrate the different 1-vs-rest scores by fitting a sigmoid [23] to them based on a validation set.

4. Experiments and results

The part-learning algorithm is evaluated on the task of scene classification on the MIT 67 indoor scene dataset of Quattoni and Torralba [24]. Note that, differently from object recognition datasets such as PASCAL VOC [8], in scene classification no geometric cue such as object bounding boxes is given to help initializing parts.

The MIT data comprises 67 indoor scene categories loosely divided into stores (*e.g.* bakery, toy store), home (*e.g.* bedroom, kitchen), public spaces (*e.g.* library, subway), leisure (*e.g.* restaurant, concert hall) and work (*e.g.* hospital, TV studio). The scenes are chosen to cover those that are best characterized by their global layout (*e.g.* corridor) and also those that are best characterized by the objects they contain (*e.g.* bookshop). Evaluation uses the protocol of [24], using the the same training and test split as [24] where each category has about 80 training images and 20 test images. In addition the training set is subdivided into about 64 train and 16 validation images. Performance is reported in terms of average classification accuracy as in [24] (*i.e.* the average of the diagonal of the confusion matrix) and, additionally, in terms of mean Average Precision (mAP).

Bag of words and Fisher vectors. The four variants of BoW and IFV (Sect. 3) are compared in Table 1) for variations in the sampling density of the RootSIFT features. Note that using RootSIFT instead of SIFT can increase the classification performance by up to 2% in some cases. The best performance is 60.77% by using the IFV encoding with a sampling step of five pixels, though the method is not very sensitive to the step size.

The part-learning method proposed by [28] had previously achieved the state-of-the-art performance on the MIT Scene 67 dataset. Their best performing achieves an accu-

Encoding	Step size: 3	Step size: 4	Step size: 5
VQ [42,000]	52.14 (51.08)	50.38 (50.50)	49.76 (50.42)
KCB [42,000]	50.59 (49.65)	49.21 (49.19)	50.41 (49.92)
LLC [42,000]	51.52 (51.87)	53.03 (51.73)	51.70 (52.09)
IFV [204,800]	60.33 (60.90)	60.67 (61.39)	**60.77** (61.05)

Table 1. **BoW scene classification results.** Performance of various BoW classifiers on the MIT Scene 67 dataset, reporting classification accuracy and mean average precision (in parentheses) for each case. The dimension of the image representation is given in square brackets. The best results significantly outperform the current state-of-the-art accuracy.

Method	Number of parts selected per class				
	10	20	30	40	50
BoP	42.34	44.81	44.96	46.00	46.10
LLC + BoP	56.66	55.98	55.93	56.01	55.94
IFV + BoP	62.80	62.75	62.65	62.02	63.10

Table 2. **Variation with number of part classifiers.** The variation of classification accuracy with number of part classifiers selected per class.

racy of 49.4% combining, in additions to the learned parts, BoW, GIST, and DPM representations. Thus it is notable that, by following the best practices indicated by [6], a solid baseline encoding is actually able to *outperform* (by 11%) the combined method of [28] as well as all other previous methods on the MIT Scene 67 dataset by using only a *single feature channel* based on RootSIFT features.

Bag of parts. Blocks are learned as described in Sect. 2. The 31-dimensional cell HOG variant of [9] is used in all the experiments. For the seeding described in Sect. 2.1, the segmentation algorithm [10] is run with parameters $k = 0.5$, $\sigma = 200$, and min $= 20$. The average number of part candidates obtained for each class is 3,800. For each of these seed blocks, a classifier is learned by following the expansion procedure of Sect. 2.2. We sample about 620,000 HOG blocks randomly from the training set, and compute the mean (μ_0) and covariance (Σ) of this set. Since Σ comes out to be low-rank and non-invertible, a regularizer ($\lambda = 0.01$) is added to the diagonal elements of the covariance matrix.

Once the parts have been learned as described in Sect. 2.2 and selected as in Sect. 2.3, the bag of parts representation is extracted from each training image as described in Sect. 3. Figure 7 shows examples of the seed blocks, the learnt HOG templates, and detections on the validation set images. Finally, 67 one-vs-rest SVMs are learned from the training images, and the resulting scene classifiers are evaluated on the test data. As one can expect, the classification accuracy increases as more parts are added to the representation (Table 2), but the peak is at around 50 parts per category. The probable reason is a lack of training material (after all the parts and classifiers are learned on the same

Method	Acc. (%)	Mean AP (%)
ROI + Gist [24]	26.05	-
MM-scene [37]	28.00	-
CENTRIST [36]	36.90	-
Object Bank [15]	37.60	-
DPM [18]	30.40	-
RBoW [19]	37.93	-
LPR [26]	44.84	-
Patches [28]	38.10	-
BoP [3,350] (Ours) (1x1)	40.31	37.31
BoP [16,750] (Ours) (1x1+2x2)	46.10	43.55
LLC [42,000] (Ours)	53.03	51.73
IFV [204,800] (Ours)	60.77	61.05

Table 3. Average classification performance of single feature methods (previous publications and this paper). The dimension of the image representation is given in square brackets. 1x1, 2x2 refers to the spatial subdivisions used.

Method	Acc. (%)	Mean AP (%)
DPM+Gist-color+SP [18]	43.10	-
Patches+GIST+SP+DPM [28]	49.40	-
LLC + BoP (Ours)	56.66	55.13
IFV + BoP (Ours)	63.10	63.18

Table 4. Average classification performance of combination of features methods (previous publications and this paper).

data) that causes overfitting. To overcome this, we left-right flip the images in the positive training set, and add them as additional positives.

Overall, the proposed part-learning method compares very favorably with the method of [28], which previously defined the state-of-the-art when using part detection on the MIT Scene 67 dataset. Our accuracy is 46.10% when 50 parts per category are used. By comparison, the accuracy of [28] is 38.10%, and they use 210 parts per category. So the parts found by our algorithm are much more informative, improving the accuracy by 8% using only a quarter of the number of detectors.

Also, our part-learning method is significantly more efficient than the discriminative clustering approach of [28] for three reasons. (i) [28] initialize their clustering algorithm by standard (generative) K-means, which, as they note, performs badly on the part clustering task; our exemplar SVM approach avoids that problem. (ii) These clusters are formed on top of a random selection of initial patches; we found that aligning seed patches to superpixels substantially increases the likelihood of capturing interesting image structures (compared to random sampling). (iii) They use *iterative hard-mining* to learn their SVM models. This approach was tested in our context and found to be 60 times slower than LDA training that avoids this step.

Combined representation. In the final experiment, the BoP (using 50 parts per class) and BoW/IFV representa-
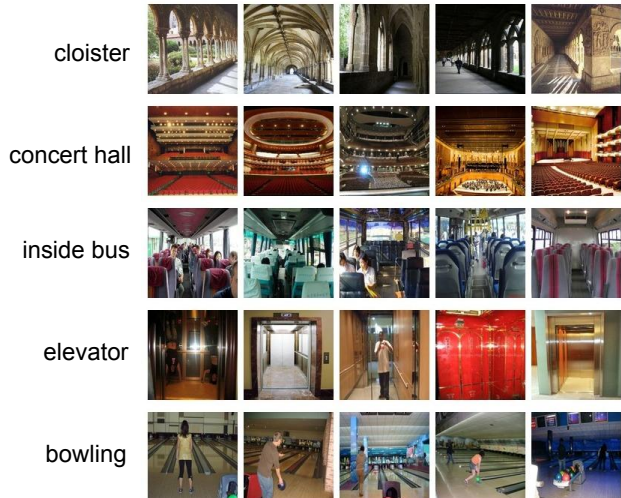

Figure 6. Categories with the highest classification rate (Combined method). Each row shows the top five results for the category

tions are combined as described in Sect. 3, and 67 one-vs-rest classifiers are learned. Table 4 reports the overall performance of the combined descriptors and compares it favorably to BoW and IFV, and hence to all previously published results. Figure 6 shows qualitative results obtained by the combined bag of parts and IFV method.

5. Summary

We have presented a novel method to learn distinctive parts of objects or scenes automatically, from image-level category labels. The key problem of simultaneously learning a part model and detecting its occurrences in the training data was solved by paced learning of Exemplar SVMs, growing a model from just one occurrence of the part. The distinctiveness of parts was measured by the new concept of entropy-rank, capturing the idea that parts are at the same time predictive of certain object categories but shareable between different categories. The learned parts have been shown to perform very well on the task of scene classification, where they improved a very solid bag of words or Fisher Vector baseline that in itself establishes the new state-of-the-art on the MIT Scene 67 benchmark.

The outcome of this work are blocks that correspond to semantically meaningful parts/objects. This mid-level representation is useful for other tasks, for example to initialize the region models of [19] or the part models of [18], and yields more understandable and diagnosable models than the original bag of visual words method.
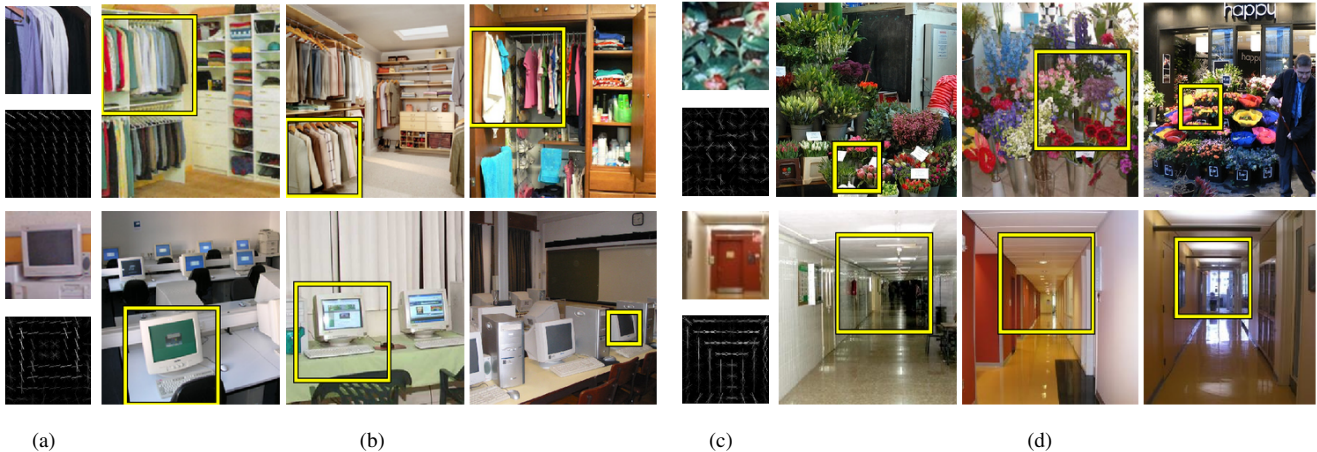
(a)   (b)   (c)   (d)

Figure 7. (a), (c) Seed blocks and the learnt HOG templates, and (b), (d) detections on the validation set images.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. In *PAMI*, 2012.

[2] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9, 1997.

[3] Y. Amit and A. Trouvé. Generative models for labeling multi-object configurations in images. In *Toward Category-Level Object Recognition*. Springer, 2006.

[4] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012.

[5] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *Proc. ICCV*, 2009.

[6] K. Chatfield, L. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proc. BMVC*, 2011.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 88(2):303–338, June 2010.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2009.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), 2004.

[11] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. In *IEEE Trans. on Computers*, 1973.

[12] M. Gharbi, T. Malisiewicz, S. Paris, and F. Durand. A gaussian approximation of feature space for fast image similarity. Technical Report 2012-032, MIT CSAIL, 2012.

[13] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *Proc. ECCV*, 2012.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006.

[15] L.-J. Li, H. Su, E. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Proc. NIPS*, 2010.

[16] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.

[17] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *Proc. ICCV*, 2011.

[18] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *Proc. ICCV*, 2011.

[19] S. Parizi, J. Oberlin, and P. Felzenszwalb. Reconfigurable models for

[20] O. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. The truth about cats and dogs. In *Proc. ICCV*, 2011.

[21] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proc. CVPR*, 2010.

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. CVPR*, 2008.

[23] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. Cambridge, 2000.

[24] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Proc. CVPR*, 2009.

[25] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *Proc. CVPR*, 2012.

[26] F. Sadeghi and M. F. Tappen. Latent pyramidal regions for recognizing scenes. In *Proc. ECCV*, 2012.

[27] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proc. ICML*, pages 807–814, 2007.

[28] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *Proc. ECCV*, 2012.

[29] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. In *Intl. Workshop on Visual Form*, 2001.

[30] K. E. A. van de Sande, J. R. R. Ujilings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *Proc. ICCV*, 2011.

[31] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *Proc. ECCV*, 2008.

[32] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proc. ICCV*, 2009.

[33] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *Proc. CVPR*, 2010.

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

[35] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *Proc. CVPR*, volume 2, pages 101–108, 2000.

[36] J. Wu and J. Rehg. Centrist: A visual descriptor for scene categorization. In *PAMI*, 2011.

[37] J. Zhu, L.-J. Li, L. Fei-Fei, and E. Xing. Large margin learning of upstream scene understanding models. In *Proc. NIPS*, 2010.

scene recognition. In *Proc. CVPR*. CVPR, 2012.