Harvesting Discriminative Meta Objects with Deep CNN Features for Scene Classification

Ruobing Wu[†][1] Baoyuan Wang[‡] Wenping Wang[†] Yizhou Yu[†]

[†] The University of Hong Kong [‡] Microsoft Technology and Research

Abstract

Recent work on scene classification still makes use of generic CNN features in a rudimentary manner. In this ICCV 2015 paper, we present a novel pipeline built upon deep CNN features to harvest discriminative visual objects and parts for scene classification. We first use a region proposal technique to generate a set of high-quality patches potentially containing objects, and apply a pre-trained CNN to extract generic deep features from these patches. Then we perform both unsupervised and weakly supervised learning to screen these patches and discover discriminative ones representing category-specific objects and parts. We further apply discriminative clustering enhanced with local CNN fine-tuning to aggregate similar objects and parts into groups, called meta objects. A scene image representation is constructed by pooling the feature response maps of all the learned meta objects at multiple spatial scales. We have confirmed that the scene image representation obtained using this new pipeline is capable of delivering state-of-the-art performance on two popular scene benchmark datasets, MIT Indoor 67 [22] and Sun397 [31].

1. Introduction

Deep convolutional neural networks (CNNs) have gained tremendous attention recently due to their great success in boosting the performance of image classification [14, 19], object detection [7, 26], action recognition [12] and many other visual computing tasks [23, 21]. In the context of scene classification, although a series of state-of-the-art results on popular benchmark datasets (MIT Indoor 67[22], SUN397 [31]) have been achieved, CNN features are still used in a rudimentary manner. For example, recent work in [33] simply trains the classical Alex's net [14] on a scene-centric dataset ("Places") and directly extracts holistic CNN features from entire images.

The architecture of CNNs suggests that they might not be best suited for classifying images, including scene images, where local features follow a complex distribution. The reason is that spatial aggregation performed by pooling layers in a CNN is too simple, and does not retain much information about local feature distributions. When critical inference happens in the fully connected layers near the top of the CNN, aggregated features fed into these layers are in fact global features that neglect local feature distributions. It has been shown in [8] that in addition to the entire image, it is consistently better to extract CNN features from multiscale local patches arranged in regular grids.

In order to build a discriminative representation based on deep CNN features for scene image classification, we need to address two technical issues: (1) Objects within scene images could exhibit dramatically different appearances, shapes, and aspect ratios. To detect diverse local objects, one could in theory add many perturbations to the input image by warping and cropping at various aspect ratios, locations, and scales, and then feed all of them to the CNN. This is, however, not feasible in practice; (2) To distinguish one scene category from another, it is much desired to harvest discriminative and representative category-specific objects and object parts. For example, to tell a "city street" from a "highway", one needs to identify objects that can only belong to a "city street" but not a "highway" scene. Pandey and Lazebnik [20] adopt the standard DPM to adaptively infer potential object parts. It is however unclear how to initialize the parts and how to efficiently learn them using CNN features.

In this paper, we present a novel pipeline built upon deep CNN features for harvesting discriminative visual objects and parts for scene classification. We first use a region proposal technique to generate a set of high-quality patches potentially containing objects [3]. We apply a pre-trained CNN to extract generic deep features from these patches. Then, for each scene category, we train a one-class SVM on all the patches generated from the images for this class as a discriminative classifier [25], which heavily prunes outliers and other non-representative patches. The

[1] This work was partially completed when the first author was an intern at Microsoft Research.

remaining patches correspond to the objects and parts that frequently occur in the images for this scene category. To further harvest the most discriminative patches, we apply a non-parametric weakly supervised learning model to screen these remaining patches according to their discriminative power across different scene categories. Instead of directly using the chosen category-specific objects and parts, we further perform discriminative clustering to aggregate similar objects and parts into groups. Each resulting group is called a **"Meta Object"**. Finally, a scene image representation is obtained by pooling the feature response maps of all the learned meta objects at multiple spatial scales to retain more information about their local spatial distribution. Locally aggregated CNN features are more discriminative than those global features fed into the fully connected layers in a single CNN.

There exists much recent work advocating the concept of middle-level objects and object parts for efficient scene image classification [16, 20, 27, 11, 4, 30]. Among them, the methods proposed in [4, 11] are most relevant. Nonetheless, there exist major differences between our method and theirs. First, we use multiscale object proposals instead of grid-based sampling with multiple patch sizes, thus we can intrinsically obtain better discriminative object candidates. Second, we aggregate our meta objects through deep CNN features while previous methods primarily rely on low-level features (i.e., HOG). As demonstrated through experiments, deep features are more semantically meaningful when used for characterizing middle-level objects. Last but not the least, there exist significantly different components along individual pipelines. For instance, we adopt unsupervised learning to prune outliers while Juneja *et al.* [11] train a large number of exemplar-SVMs, which is more computationally intensive. Furthermore, our discriminative clustering component also plays an important role in aggregating meta objects.

In summary, this paper has the following contributions: (1) We propose a novel pipeline for scene classification that is built on top of deep CNN features. The advantages of this pipeline are orthogonal to any category independent region proposal methods [29, 34, 3] and middle-level parts learning algorithms [4, 20, 11]. (2) We propose a simple yet efficient method that integrates unsupervised and weakly supervised learning for harvesting discriminative and representative category-specific patches, which we further aggregate into a compact set of groups, called meta objects, via discriminative clustering. (3) Instead of global fine-tuning, we locally fine-tune the CNN using the meta objects discovered from the target dataset. We have confirmed through experiments that the scene image representation obtained using this pipeline is capable of delivering state-of-the-art performance on two popular scene benchmark datasets, MIT Indoor 67 [22] and Sun397 [31].

2. A New Pipeline for Scene Classification

In this section, let us present the main components of our proposed new pipeline for scene classification. As illustrated in Figure 1, our pipeline is built on top of a pre-trained deep convolutional neural network, which is regarded as a generic feature extractor for image patches. In the context of scene classification, instead of directly transferring these features [33] or global fine-tuning on whole images using the groundtruth labels [6, 7], we perform local fine-tuning on discriminative yet representative local patches that correspond to visual objects or their parts. As for scene classification datasets, bounding boxes or segment masks are not available for our desired local patches. In order to harvest them, we first adapt the latest algorithms to generate image regions potentially containing objects, expecting a high recall of all informative ones (Section 2.1). Then we first apply an unsupervised learning technique, one-class SVMs, to prune those proposed regions that do not appear frequently in the images for a specific scene class. This is followed by a weakly supervised learning step to screen the remaining region proposals and discard those patches that are unlikely to be useful for differentiating a specific scene category from other categories (Section 2.2).

To further improve the generality and representativeness of the remaining patches, we perform discriminative clustering to aggregate them into a set of meta objects (Section 2.3). Finally, our scene image representation is built on top of the probability distribution of the mined meta objects (Section 2.4).

2.1. Region Proposal Generation

As discussed in Section 1, for arbitrary objects with varying size and aspect ratio, the traditional sliding window based object detection paradigm requires multiresolution scanning using windows with different aspect ratios. For example, in pedestrian detection [5], at least two windows should be used to search for the full body and upper body of pedestrians. Recently, an alternative paradigm has been developed that performs perceptual grouping with the goal of proposing a limited number of high-quality regions, that likely enclose objects. Tasks including object detection [7] and recognition [9] can then be built on top of these proposed regions only without considering other non-object regions. There is a large body of literature along this new paradigm for efficiently generating region proposals with a high recall, including selective search [29], edge-boxes [34], and multi-scale combinatorial grouping (MCG) [3]. We empirically choose MCG as the first component in our pipeline for generating high-quality region proposals, but one can use other methods as well. Figure 3 shows a few examples of regions generated by MCG. We also use region proposals from hierarchical image segmentation [2] at the same time (see Sec.2.5).

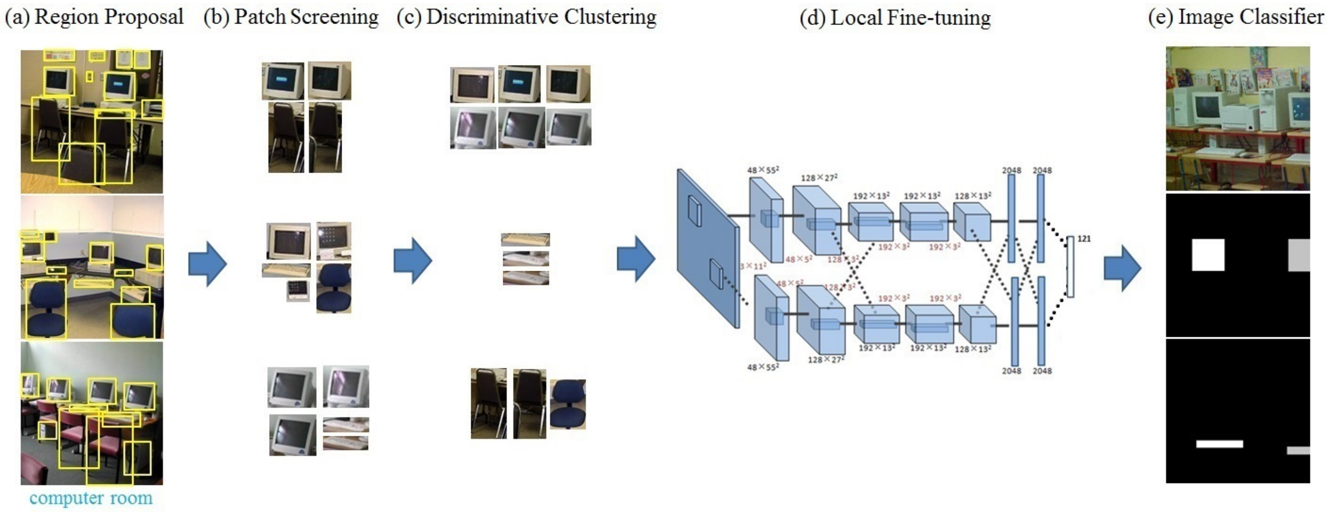

Figure 1. Flowchart of our pipeline. From left to right: (a) Training scene images are processed by MCG [3] and we obtain top ranked region proposals (yellow boxes). (b) Patches are screened by our non-parametric scheme and only discriminative patches remain. (c) Discriminative clustering is performed to build meta objects. Three meta objects are shown here: 'computer screen', 'keyboard', 'computer chair' (from top to bottom). Note that these names are for demonstration only, not labels applicable to our pipeline. (d) Local fine-tuning is performed on Hybrid CNN [33], which decides which meta object a testing region belongs to. (e) We train an image classifier on aggregated responses of our fine-tuned CNN. Here the response maps of two meta objects, "computer screen" (second row) and "keyboard" (bottom row), are shown. Gray-scale values in the response maps indicate confidence.

Feature Extraction We use the CNN model pre-trained on the Places dataset [33] as our generic feature extractor for all the image regions generated by MCG. As this CNN model only takes input images with a fixed resolution, we follow the warping scheme described in R-CNN [7] and resample a patch with an arbitrary size and aspect ratio using the required resolution. Then each patch propagates through all the layers in the pre-trained CNN model, and we take the 4096-dimensional vector in the FC7 layer as the feature representation of the patch (see [14] and [33] for detailed information about the network architecture).

2.2. Patch Screening

Screening via One-Class SVMs For each scene category, there typically exist a set of representative regions that frequently appear in the images for that category. For example, since regions with computer monitors frequently appear in the images for the "computer room" class, a region containing monitors should be a representative region. Meanwhile, there are other regions that might only appear in few images. Such non-representative patches can be viewed as outliers for a certain scene category. On the basis of this observation, we adopt one-class SVMs [25] as discriminative models for removing non-representative patches. A one-class SVM separates all the data samples from the origin to achieve outlier detection. Let $x_1, x_2, ..., x_l (x_i \in R^d)$ be the proposed regions from the same class, and $\Phi : X \longrightarrow H$ be a kernel function that maps original region features into another feature space. Training a one-class SVM needs to solve the following optimization:

$$\min_{w,\xi,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{\upsilon l}\sum_{i=1}^{l}\xi_i - \rho \tag{1}$$

subject to

$$(w \cdot \Phi(x_i)) \geq \rho - \xi_i,\ \xi_i \geq 0,\ i = 1, 2, ..., l,$$

where $\upsilon (\in (0, 1])$ controls the ratio of outliers. The decision function

$$f(x) = \text{sign}(w \cdot \Phi(x_i) - \rho) \tag{2}$$

should return the positive sign given the representative patches and the negative sign given the outliers. This is because the representative patches tend to stay in a local region in the feature space while the outliers are scattered around in this space. To further improve the performance, we train a series of cascaded classifiers, each of which labels 15% of the input patches as outliers and prune them. We typically use 3 cascaded classifiers.

Weakly Supervised Soft Screening After the region proposal step and outliers removal, let us suppose that m_i image patches have been generated for each image I_i, and these patches likely contain objects or object parts. Let us denote a patch from I_i as p^i_j $(j \in \{1, ..., m_i\})$, and use y_i to represent the scene category label of image I_i. We associate each image patch p^i_j with a weight $w^i_j \in [0, 1]$ indicating the discriminative power of the patch among scene

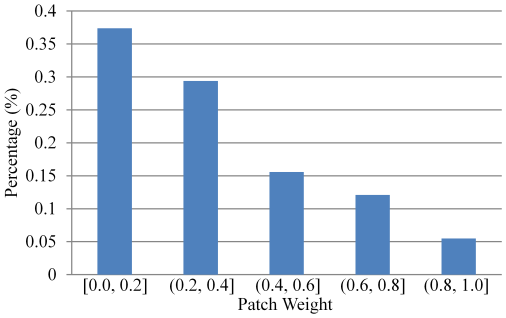

Figure 2. Patch weight distribution after weakly supervised patch screening.

category labels. Our goal is to estimate this weight for every patch. Intuitively, a discriminative patch should have a high probability of appearing in one scene category and low probabilities of appearing in the other categories. That means, if we find the set of K nearest neighbors N_j^i of p_j^i from all image patches generated from all training images except I_i, we can use the following class density estimator to set w_j^i:

$$w_j^i = P(y_i|p_j^i) = \frac{P(p_j^i, y_i)}{P(p_j^i)} \approx K_y/K, \tag{3}$$

where K_y is the number of patches among the K nearest neighbors that share the same scene label with p_j^i. By assuming that the K nearest neighbors of p_j^i are almost identical to p_j^i, we use K_y to estimate the joint probability between a patch p_j^i and its label y_i. Empirically we set K to 100 in all the experiments. It is worth noting that patches with large weights also have more representative power. As representative patches would occur frequently in the visual world [27], it is unlikely for non-representative patches to find similar ones (as its nearest neighbors) that share the same scene label. Fig. 2 shows the distribution of patch weights after our screening process.

2.3. Meta Object Creation and Classification

Once we have identified the most discriminative image patches, the next step is grouping these patches into clusters such that ideally patches in the same cluster should contain visual objects that belong to the same category and share the same semantic meaning. This is important for discovering the relationship between scene category labels and the labels of object clusters. Clustering also helps to show the internal variation of an object label. For example, desks facing a few different directions in a classroom might be grouped into several clusters. We call every patch cluster a meta object. Note that meta objects could correspond to visual objects but could also correspond to parts and patches that characterize the commonalities within a scene category.

We adopt the Regularized Information Maximization (RIM) algorithm [13] to perform discriminative clustering. RIM strikes a balance among cluster separation, cluster balance and cluster complexity. Fig. 3 shows a few clusters after applying RIM to the screened discriminative patches from the MIT 67 Indoor Scenes dataset [22]. As we can see, the patches within the same cluster has similar appearances and the same semantic meaning. Here we can also observe the discriminative power of such clusters. For example, the wine buckets (top row in Fig. 3) only show up in wine cellars, and the cribs (second row from the bottom in Fig. 3) only show up in nurseries.

Local Fine-Tuning for Patch Classification Given the set of meta objects, we need a classifier to decide which meta object a patch from a testing image belongs to. There are various options for this classifier, including GMM-type probabilistic models, SVMs, and neural networks. We choose to fine-tune the pre-trained CNN on our meta objects, which include the collection of discriminative patches surviving the patch screening process. We perform stochastic gradient descent over the pre-trained CNN using the warped discriminative image patches and their corresponding meta object labels. Take MIT Indoor 67 [22] as an example. After weakly supervised patch screening (Section 2.2), there exist around a million remaining image patches, and 120 meta objects are discovered during the clustering step (Section 2.3). In the CNN, we replace the original output layer that performs ImageNet-specific 1000-way classification with a new output layer that does 121-way classification while leaving all other layers unchanged. Note that we need to add one extra class to represent those patches that are discarded during the screening step. The reason for local fine-tuning is obtaining an accurate meta object classifier that is also robust to noisy labels generated by the discriminative clustering algorithm used in Section 2.3.

2.4. Image Representation with Meta Objects

Inspired by previous work such as object-bank [16] and bag-of-parts (BOF) [11], we hypothesize that any scene image can be represented as a bag of meta objects as well. Suppose N meta objects have been learned during discriminative clustering (Section 2.3).

Given a testing image, we still perform MCG to obtain region proposals. Every region can be classified into one of the discriminative object clusters using our meta object classifier. Spatial aggregation of these meta objects can be performed using Spatial Pyramid Matching (SPM) [32]. In our implementation, we use three levels of SPM, and adaptively choose the centroid of all meta objects falling into a SPM region as the splitting center of its subregions. This strategy can better balance the number of meta objects that fall into each subregion. After applying SPM to the testing

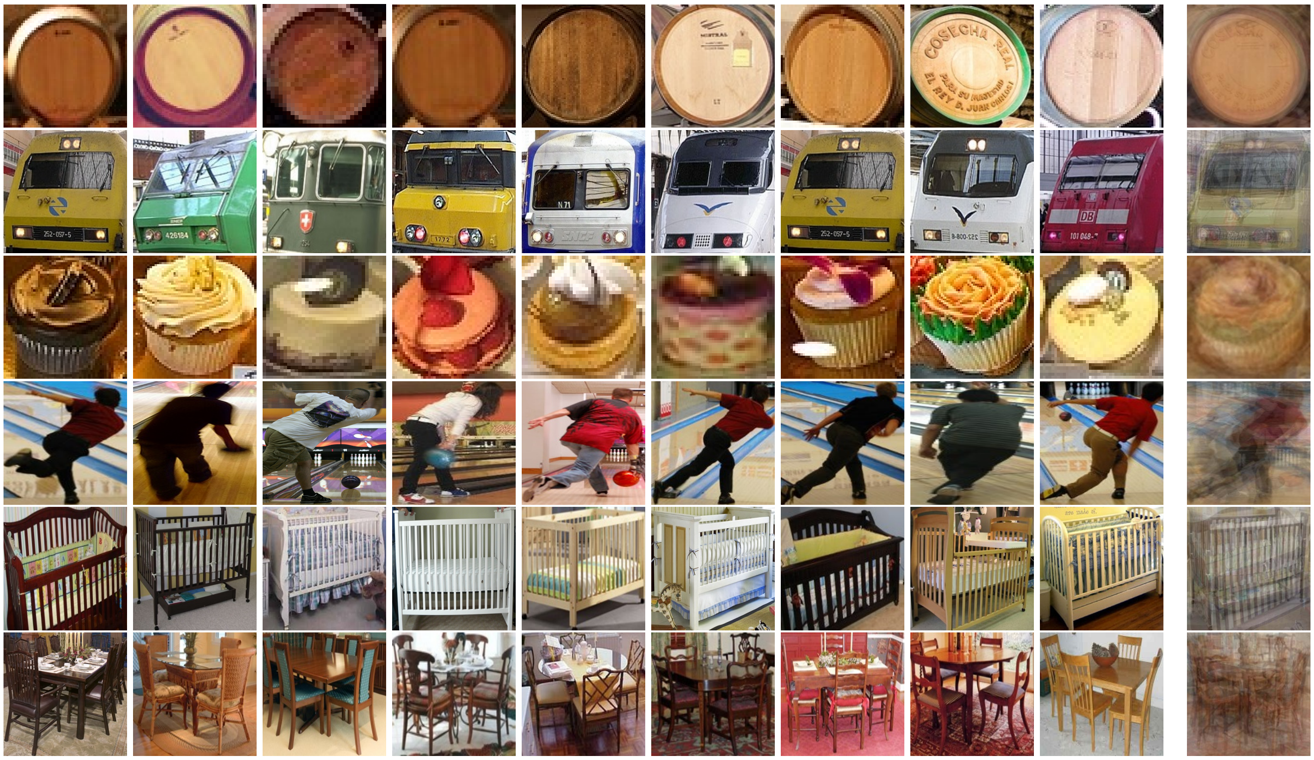

Figure 3. Examples of patch clusters (meta objects) from the MIT 67 Indoor dataset [22]. Patches on the same row belong to the same meta object. The rightmost column shows the average image, namely the 'center', of the corresponding meta object.

image, we obtain a hierarchical spatial histogram of meta object labels over the image, which can be used for determining the scene category of this testing image.

Another pooling method we consider is Vector of Locally Aggregated Descriptors (VLAD) [10, 1]. We compute a modified version of VLAD that suits our framework. Specifically, we use our discriminative object clusters (meta objects) as the clusters for computing VLAD. That means we do not perform K-means clustering for VLAD. It is important to maintain such consistency because otherwise the recognition performance would degrade by 1.5% on the MIT 67 Indoor Scenes dataset (from 78.41% to 76.9%). Other steps are similar to the standard VLAD. Given region proposals of an image, we assign each region to its nearest cluster center, and aggregate the residuals of the region features, resulting in a 4096-d vector per cluster. Suppose there are k clusters. The dimension of this per-cluster vector is reduced to (4096/k)-d using PCA. Finally, these (4096/k)-d vectors are concatenated into a 4096-d VLAD descriptor.

The holistic Places CNN feature extracted from the whole image is also useful for training the scene image classifier since they also encode local as well as global information of the scene.

We train a neural network with two fully-connected hidden layers (each with 200 nodes) using normalized VLAD (or SPM) features concatenated with the holistic Places CNN features. The relative weight between these two types of features are learned via cross validation on a small portion of the training data. We use the rectified linear function (ReLU) as the activation function of the neurons in the hidden layers.

2.5. Multi-Level Image Representation

Our image representation with meta objects can be generalized to a multi-level representation. The insight here is that objects with different sizes and scales may supply complementary cues for scene classification. To achieve this, we switch to multi-level region proposals. The coarser levels deal with larger objects, while the finer levels deal with smaller objects and object parts. On each level, region proposals are generated and screened separately. Local fine-tuning for patch classification is also performed on each level separately. During the training stage of the final image classifier, the image representation is defined as the concatenation of the feature vectors from all levels. In practice, we find a 2-level representation sufficient. The bottom level includes relatively small regions from a finer level in a region hierarchy [2] to capture small objects and object parts in an image, while the top level includes region proposals generated by MCG as well as relatively large regions from a coarser level in the region hierarchy to capture large objects.

3. Experiments and Discussions

In this section, we evaluate the performance of our framework, named MetaObject-CNN, on the MIT Indoor 67 [22] and SUN 397 [31] datasets as well as analyze the effectiveness of the specific choices we made at every stage of our pipeline introduced in Section 2.

3.1. Datasets

MIT Indoor 67 MIT Indoor 67 [22] is a challenging indoor scene dataset, which contains 67 scene categories and a total of 15,620 images. The number of images varies across categories (but at least 100 images per category). Indoor scenes tend to have more variations in terms of composition, and are better characterized by the objects they have. This is consistent with the motivation of our framework.

SUN397 SUN397 [31] is a large-scale scene dataset, which contains 397 scene categories and a total of 108,754 images (also at least 100 images per category). The categories include different kinds of indoor and outdoor scenes which show tremendous object and alignment variance, thus bringing more complexity in learning a good classifier.

3.2. Experimental Setup

For MIT Indoor 67, we train our model on the commonly adopted benchmark, which contains 80 training images and 20 testing images per category. There are 192 top ranked region proposals generated with MCG and 32 (96) regions from hierarchical image segmentation in the top (bottom) level for every training and testing image. The feature representation of a proposed region is set to the 4096-dimensional vector at the FC7 layer of the Hybrid CNN from [33]. After outlier removal (3 iterations of 15% filtering out), we further discard 16% patches, where the ratio is determined via cross validation on a small portion of the training data. Then we perform data augmentation (to 4 times larger) on the remaining patches using reflection, small rotation and random distortion. Discriminative clustering is performed on the augmented patches to produce 120 (40) meta objects for local fine-tuning in the bottom (top) level, which is performed on the Hybrid CNN by replacing the original output layer that performs ImageNet-specific 1000-way classification with a new output layer that does 121-way (41-way) classification while leaving all other layers unchanged. The pooling step (SPM and our modified VLAD) is discussed in Section 2.4. The image classification is done by a neural network with two fully-connected layers (200 nodes each) on the concatenated feature vector of VLAD pooling and the Hybrid CNN feature of the whole image.

For SUN 397, we adopt the commonly used evaluation benchmark that contains 50 training images and 50 testing images per category for each split from [31]. There are 96 top ranked regions generated with MCG and 32 (96) regions from hierarchical image segmentation in the top (bottom) level for every training and testing image. The feature representation of a proposed region is also set to the 4096-dimensional vector at the FC7 layer of the Hybrid CNN. After outlier removal (3 iterations of 15% filtering out), we further discard 24% patches. Data augmentation is also performed on the remaining patches involving reflection, small rotation and random distortion. Discriminative clustering results in 450 (150) meta objects in the bottom (top) level. Local fine-tuning is further performed on the Hybrid CNN by replacing the original output layer with a new output layer that does 451-way (151-way) classification while leaving all other layers unchanged. We also train a neural network with two fully-connected layers (200 nodes each) on the concatenated feature vector of VLAD pooling and the Hybrid CNN feature of the whole image to deal with image level classification.

3.3. Comparisons with State-of-the-Art Methods

In Table 1, we compare the recognition rate of our method (MetaObject-CNN) against published results achieved with existing state-of-the-art methods on MIT Indoor 67. Among the existing methods, oriented texture curves (OTC) [18], spatial pyramid matching (SPM) [15], and Fisher vector (FV) with bag of parts [11] represent effective feature descriptors as well as their associated pooling schemes. Discriminative patches [27, 4] are focused on mid-level features and representations. More recently, deep learning and deep features have proven to be valuable to scene classification as well [8, 33]. The recognition accuracy of our method outperforms the state of the art by around 8.1%.

Table 1. Scene Classification Performance on MIT Indoor 67

Method	Accuracy(%)
SPM [15]	34.40
OTC [18]	47.33
Discriminative Patches ++ [27]	49.40
FV + Bag of parts [11]	63.18
Mid-level Elements [4]	66.87
MOP-CNN [8]	68.88
Places-CNN [33]	68.24
Hybrid-CNN [33]	70.80
MetaObject-CNN	**78.90**

Table. 2 shows a comparison between the recognition rate achieved with our method (MetaObject-CNN) and those achieved with existing state-of-the-art methods on the SUN397 dataset. In addition to the methods introduced earlier, there exists additional representative work here. Xiao *et al.* [33], as the collector of SUN397, integrated 14 types of distance kernels including bag of features and GIST. DeCAF [6] uses the global 4096D feature from a pre-trained CNN model on ImageNet. OTC together with the HOG2x2 descriptor [18] outperforms dense Fisher vectors [24], both of which are effective feature descriptors for SUN397. And again, by applying deep learning techniques, MOP-CNN [8] and Places-CNN [33] (fine-tuned on SUN397) achieve state-of-the-art results (51.98% and 56.2%). With our MetaObject-CNN pipeline, we manage to achieve a higher recognition accuracy.

Table 2. Scene Classification Performance on SUN397

Method	Accuracy(%)
OTC [18]	34.56
Xiao et al. [33]	38.00
DeCAF [6]	40.94
FV [24]	47.20
OTC+HOG2x2 [18]	49.60
MOP-CNN [8]	51.98
Hybrid-CNN [33]	53.86
Places-CNN [33]	56.20
MetaObject-CNN	**58.11**

3.4. Evaluation and Discussion

In this section, we perform an ablation study to analyze the effectiveness of individual components in our pipeline. When validating each single component, we keep all the others fixed. Specifically, we treat the final result from our MetaObject-CNN as the baseline, and perform the analysis by altering only one component at a time. Table 3 shows a summary of the comparison results on MIT Indoor 67. A detailed explanation of these results is given in the rest of this section.

Table 3. Evaluation results on MIT Indoor 67 for varying pipeline configurations.

Configuration	Accuracy(%)
Global fine-tuning	73.88
Mode-seeking [4] with Hybrid-CNN	69.70
Mode-seeking elements instead of MCG	76.34
Dense grid-based patches	71.43
Without outlier removal and patch screening	75.12
Without outlier removal	76.30
Without patch screening	78.82
Without clustering	72.81
Without local fine-tuning	76.10
Cross-dataset evaluation	76.52
MetaObject-CNN	**78.90**

Global vs Local Fine-Tuning Most of the previous methods [33, 6, 12] using a pre-trained deep network primarily focus on global fine-tuning for domain adaptation tasks, which take the entire image as input and rely on the network itself to learn all the informative structures embedded within a new dataset. However, in this work, we perform fine-tuning on local meta objects harvested in an explicit manner. To compare, we start with the Places CNN network [33], and fine-tune this network on MIT Indoor 67. The recognition rate after such global fine-tuning is 73.88% (top row in Table. 3), which is around 5% lower than that of our pipeline. This indicates the advantages of our local approach of harvesting meta objects and performing recognition on top of them.

Choice of Region Proposal Method In addition to choosing MCG [3] and hierarchical image segmentation for generating object proposals, one might directly use dense grid-based patches or mid-level discriminative patches discovered by the pioneering techniques in [4, 27] as local object proposals. To evaluate the effectiveness of MCG, we have conducted the following three internal comparisons.

First, we compare our patch screening on top of region proposal with the patch discovery process in [4], which is a piece of representative work on learning mid-level patches in a supervised manner. For a fair comparison, we use the Places CNN feature (FC7) to represent the visual elements in this work. Similar to the configuration in [4], 1600 elements are learned per class and 200 top elements per class are used for further classification. The resulting recognition rate is 69.70% (second row in Table. 3, which is 9.2% lower than our result. This comparison demonstrates that region proposal plus patch screening is helpful in finding visual objects that characterize scenes. In a second experiment, we feed the top visual elements identified by [4] to our patch clustering step, and obtain 96 meta objects. The final recognition rate achieved with these meta objects is 76.34% (third row in Table. 3), which is around 2.6% lower than our result. This second experiment shows that MCG works with our pipeline better than mode-seeking elements from [4]. Then in a third experiment, instead of taking region proposals, we have tried using all patches from a regular 8x8 grid, the result is 71.43% (fourth row in Table. 3), which indicates patches sampled from a regular grid are not good candidates for meta objects.

Importance of Outlier Removal and Patch Screening To see how important our outlier removal and patch screening stages are, one can directly feed all the object proposals without any screening into the subsequent components down the pipeline (discriminative clustering and local fine-tuning). During our patch screening step, as shown in Eq. 3, we rank all the patches according to their discriminative weights and discard those with lower weights. Here we define the total screening ratio as the percentage of discarded patches in both outlier removal and patch soft screening steps. In Fig. 4 (top), we can see, when the total screening ratio is zero, the recognition accuracy is 75.12% (also shown in the fifth row in Table. 3). This is because, although we have reasonable region proposals, there could still be many noisy ones among them. These noisy region proposals are either false positives or non-discriminative objects (as shown in Fig. 1) shared by multiple scene categories. On the other hand, an overly high screening ratio has also been found to hurt recognition performance, as shown in Fig. 4 (top). This is reasonably easy to understand because higher ratios could discard some discriminative meta objects that would otherwise contribute to the overall performance. We search for an optimal ratio through cross validation on a small subset of the training data. The outlier removal step is also important in filtering out regions that do not fit in a certain category and brings along 2.6% improvement in final classification performance, as shown in the sixth row of Table. 3).

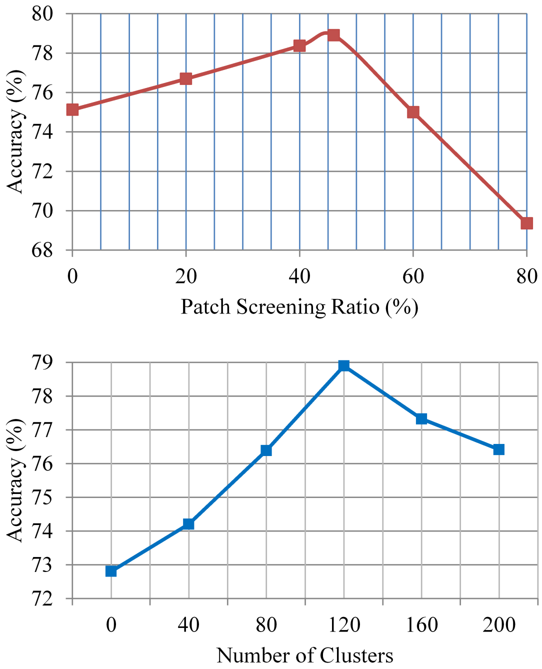

Figure 4. Top: recognition accuracy vs. total screening ratio on MIT Indoor 67. Bottom: recognition accuracy vs. number of clusters in bottom level on MIT Indoor 67.

Importance of Clustering Next we justify the usefulness of clustering patches into meta objects. Without patch clustering, we can directly take the collection of screened patches as a large codebook, and treat every patch as a visual word. We then apply LSAQ [17] (with 100 nearest neighbors) coding and SPM pooling to build the image-level representation. The resulting recognition rate on MIT Indoor 67 is 72.81% (eighth row in Table. 3), which is around 6.1% lower than the result of MetaObject-CNN. This controlled experiment demonstrates that patch clustering for meta object creation is crucial in our pipeline. Clustering patches into meta objects improves the generality and representativeness of those discovered discriminative patches because clustering emphasizes the common semantic meaning shared among similar patches while tolerating less important differences among them. Fig. 4 (bottom) shows the impact of the number of clusters in the bottom level on the final recognition rate. It is risky to group patches into an overly small number of clusters because it would assign patches with different semantic meanings to the same meta object. Creating too many clusters is also risky due to the poor generality of the semantic meanings of meta objects.

Importance of Local Fine-Tuning Local fine-tuning has also proven to be effective in our pipeline. We tried using the responses from the RIM clustering model directly for pooling. On MIT Indoor 67, the recognition rate without local fine-tuning is 76.10% (ninth row from the bottom in Table. 3), which is around 2.8% lower than that with local fine-tuning. This demonstrates local fine-tuning actually defines better separation boundaries between clusters, which is consistent with the common sense about fine-tuning. We have also used the CNN locally fine-tuned on SUN397 to perform cross-dataset classification on MIT Indoor 67. The recognition rate is 76.52% (bottom row in Table. 3), which indicates CNNs fine-tuned over one scene patch dataset have the potential to perform well on other scene datasets.

4. Conclusions

We have introduced a novel pipeline for scene classification, which is built on top of pre-trained CNN networks via explicitly harvesting discriminative meta objects in a local manner. Through extensive comparisons in a series of controlled experiments, our method generates state-of-the-art results on two popular yet challenging datasets, MIT Indoor 67 and Sun397. Recent studies on convolutional neural networks, such as GoogLeNet [28], indicate that using deeper models would improve recognition performance more substantially than shallow ones. Therefore training better generic CNNs would certainly improve its transfer learning capability as well. Nevertheless, our approach is intrinsically orthogonal to this line of effort. Exploring other local fine-tuning methods would be an interesting direction for future work.

Acknowledgment This work was partially supported by Hong Kong Research Grants Council under General Research Funds (HKU17209714).

References

[1] R. Arandjelovic and A. Zisserman. All about vlad. In *CVPR*, pages 1578–1585. IEEE, 2013. 5

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011. 2, 5

[3] P. A. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marqués, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, pages 328–335, 2014. 1, 2, 3, 7

[4] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS'13*, 2013. 2, 6, 7

[5] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *TPAMI*, 34(4):743–761, April 2012. 2

[6] 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. In *ICML*, pages 647–655, 2014. 2, 7

[7] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2, 3

[8] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014. 1, 6, 7

[9] C. Gu, J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1030–1037, June 2009. 2

[10] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311. IEEE, 2010. 5

[11] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013. 2, 4, 6

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 1, 7

[13] A. Krause, P. Perona, and R. G. Gomes. Discriminative clustering by regularized information maximization. In *NIPS*, 2010. 4

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 1, 3

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 6

[16] L. Li, H. Su, Y. Lim, and F. Li. Object bank: An object-level image representation for high-level visual recognition. *IJCV*, 107(1):20–39, 2014. 2, 4

[17] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *ICCV*, 2011. 8

[18] R. Margolin, L. Zelnik-Manor, and A. Tal. Otc: A novel local descriptor for scene classification. In *ECCV 2014*, pages 377–391. Springer, 2014. 6, 7

[19] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 1

[20] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 1, 2

[21] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *ACCV*, 2014. 1

[22] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, pages 413–420, 2009. 1, 2, 4, 5, 6

[23] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, 2014. 1

[24] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013. 7

[25] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, July 2001. 1, 3

[26] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, 2013. 1

[27] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV 2012*, pages 73–86, 2012. 2, 4, 6, 7

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, 2014. 8

[29] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. 2

[30] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu. Max-margin multiple-instance dictionary learning. In *ICML-13*, volume 28, May 2013. 2

[31] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR'10*, June 2010. 1, 2, 6

[32] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801. IEEE, 2009. 4

[33] B. Zhou, J. Xiao, A. Lapedriza, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 1, 2, 3, 6, 7

[34] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 2