Food-101 – Mining Discriminative Components with Random Forests

Lukas Bossard[1] Matthieu Guillaumin[1] Luc Van Gool[1,2]

[1]Computer Vision Lab, ETH Zürich, Switzerland
lastname@vision.ee.ethz.ch
[2]ESAT, PSI-VISICS, K.U. Leuven, Belgium
vangool@esat.kuleuven.be

Abstract. In this paper we address the problem of automatically recognizing pictured dishes. To this end, we introduce a novel method to mine discriminative parts using Random Forests (RF), which allows us to mine for parts simultaneously for all classes and to share knowledge among them. To improve efficiency of mining and classification, we only consider patches that are aligned with image superpixels, which we call components. To measure the performance of our RF component mining for food recognition, we introduce a novel and challenging dataset of 101 food categories, with 101'000 images. With an average accuracy of 50.76%, our model outperforms alternative classification methods except for CNN, including SVM classification on Improved Fisher Vectors and existing discriminative part-mining algorithms by 11.88% and 8.13%, respectively. On the challenging MIT-Indoor dataset, our method compares nicely to other s-o-a component-based classification methods.

Keywords: Image classification, Discriminative part mining, Random Forest, Food recognition

1 Introduction

Food is an important part of everyday life. This clearly ripples through into digital life, as illustrated by the abundance of food photography in social networks, dedicated photo sharing sites and mobile applications.[1] Automatic recognition of dishes would not only help users effortlessly organize their extensive photo collections but would also help online photo repositories make their content more accessible. Additionally, mobile food photography is now used to help patients estimate and track their daily calory intake, outside of any constraining clinical environment. However, current systems resort to nutrition experts [27] or Amazon Mechanical Turk [30] to label food items.

Despite these numerous applications, the problem of recognizing dishes and the composition of their ingredients has not been fully addressed by the computer vision community. This is not due to the lack of challenges. In contrast to scene

[1] *E.g.*: foodspotting.com, sharedappetite.com, foodgawker.com, *etc.*

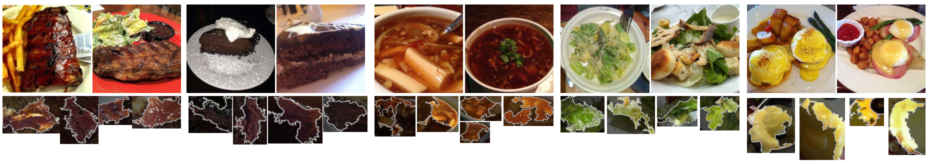


Fig. 1: Typical examples of our dataset and corresponding mined components. From left to right: baby back ribs, chocolate cake, hot and sour soup, caesar salad, eggs benedict. [All our figures are best viewed in color]

classification or object detection, food typically does not exhibit any distinctive spatial layout: while we can decompose an outdoor scene with a ground plane, a horizon and a sky region, or a human as a trunk with a head and limbs, we cannot find similar patterns relating ingredients of a mixed salad. The point of view, the lighting conditions, but also (and not least) the very realization of a recipe are among the sources of high intra-class variations. On the bright side, the nature of dishes is often defined by the different colors and textures of its different local components, such that humans can identify them reasonably well from a single image, regardless of the above variations. Hence, food recognition is a specific classification problem calling for models that can exploit local information.

As a consequence, we aim at identifying discriminative image regions which help distinguish each type of dish from the others. We refer to those as components and show a few examples in Fig. 1. To mine for such components, we introduce a weakly-supervised mining method which relies on Random Forests [14,4]. It is similar in spirit to previously proposed mid-level discriminative patch mining work [7,35,38,25,8,19,34,40]. Our Random Forest mining framework differs from all these works in the following points: First, it mines for discriminative components simultaneously for all classes, compared to independently. This speeds up the training process and allows to share knowledge between classes. Second, we restrict the search space for discriminative parts to patches aligned with superpixels, instead of sampling random image patches, in a spirit similar to what has been successfully proposed in the context of object detection [36,12]. As a consequence, not only do we manipulate regions that are consistent in color and texture, but we can afford extracting stronger visual features to improve classification. This also dramatically reduces the classification complexity on test images as the numbers of component classifiers/detectors can be fairly large (hundreds to several ten thousands): we typically use only a few dozens of superpixels per image, compared to tens of thousands of sliding windows.

The paper also introduces a new, publicly available dataset for real-world food recognition with 101'000 images. We coin this dataset *Food-101*, as it consists of 101 categories. To the best of our knowledge, this is the first public database of its kind. So far, research on food recognition has been either performed on closed, proprietary datasets [15] or on small-scale image sets taken in a controlled laboratory environment [5,39].

In summary, this paper makes the following contributions: (i) A novel discriminative part mining method based on Random Forests. (ii) A superpixel-based patch sampling strategy that prevents running many detectors on sliding windows. (iii) A novel, large scale and publicly available dataset for food recognition. (iv) Experiments showing that our approach outperforms the state-of-the-art Improved Fisher Vectors classifier [32] and the part-based mining approach of [34] on Food-101. On the MIT-Indoor dataset, our method compares nicely to very recent mining methods and is competitive with IFV.

We discuss related work in the next section. Our novel dataset is described in Section 3. In Section 4, we introduce our component mining and classification framework. Our method is evaluated in Section 5, and we conclude in Section 6.

2 Related Work

Image classification is a core problem for computer vision, with many recent advances coming from object recognition. Classical approaches exploit interest point descriptors, extracted locally or on a dense grid, then pooled into a vectorial representation to use SVM for classification. Recent advances highlight the importance of nonlinear feature encoding, *e.g.*, Fisher Vectors [32] and spatial pooling [24]. A very recent and successful trend in classification is to try and identify discriminative object (or scene) parts (or patches) [7,35,38,25,8,19,34,40], drawing on the success of deformable part-based models (DPM) for object detection [9]. This can consist of (a) finding prototypes for regions of interest [31,40], (b) mining patches whose associated binary SVM obtains good classification accuracy on a validation set [34], (c) clustering patches with a multi-instance SVM (MI-SVM) [38] on a external dataset [25], (d) optimizing part detectors in a latent SVM framework [35], (e) evaluating many exemplar-SVMs [8,19] on sliding windows, exploiting discriminative decorrelation [13] to speed-up the process, or (f) identifying discriminative modes in the HOG feature space [7].

While this work represents a variant of discriminative part mining, it differs in various ways from previous work. In contrast to all other discriminative part mining methods, we efficiently and simultaneously mine for discriminative parts for all the categories in our dataset thanks to the multi-class nature of Random Forests. Secondly, while all other methods employ a computationally expensive (often multi scale) sliding window detection approach to produce the part score maps for the final classification step, our approach employs a simple yet effective window selection by exploiting image superpixels.

Concerning food recognition, most works follow a classical recognition pipeline, focusing on feature combination and on specialized datasets. [18] uses a private dataset of Japanese food, later augmented with more features and classes [20]. Similarly, [6] jointly classifies and estimates quantity of 50 Chinese food categories using private data. [28] uses DPM to locally pool features. Food images obtained in a controlled environment are also popular in the literature. The Pittsburgh food dataset [5] contains 101 classes, but with only 3 instances per class and 8 images per instance. Yang *et al.* [39] propose to learn spatial re-

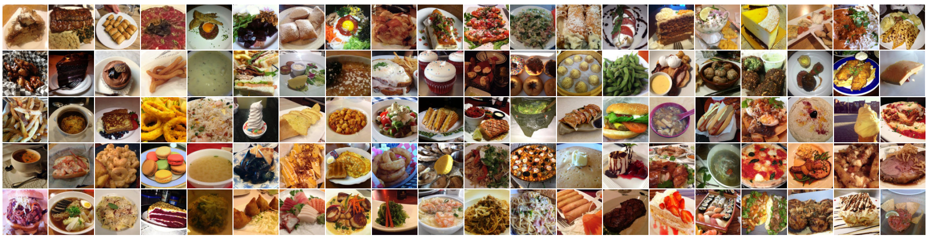


Fig. 2: Here we show one example for 100 out of the 101 classes in our dataset. Note the high variance in food type, color, exposure and level of detail, but also visually and semantically similar food types.

lationships between ingredients using pairwise features. This approach is bound to work only for standardized meals.

We resort to Random Forests (RF) [14,4] for mining discriminative regions in images. They are a well-established clustering and classification framework and proved successful for many vision applications, including object recognition [3,29,33], object detection [11] and semantic segmentation [22,33]. Our use of RF is different compared to those works. Instead of directly using a RF for classification of patches [3] or learning specific locations of interest in images [40], we are using RF to discriminatively cluster superpixels into groups (leaves), and then use the leaf statistics to select the most promising groups (*i.e.*, mine for parts). For this key step, we have developed a *distinctiveness* measure for leaves, and ensure that distinctive but near-duplicate leaves are merged. Once parts are mined, the RF is entirely discarded and is not used at classification time (in contrast to [3,29,40]). Instead we model the mined components explicitly and directly using SVMs. At test time, only those SVM need to be evaluated on the image regions.

3 Dataset: Food-101

As noted above, to date, only the PFID dataset [5] is publicly available. However, it contains only standardized fast food images taken under laboratory conditions. Therefore, we have collected a novel real-world food dataset by downloading images from foodspotting.com. The site allows users to take images of what they are eating, annotate place and type of food and upload these information online. We chose the top 101 most popular and consistently named dishes and randomly sampled 750 training images. Additionally, 250 test images were collected for each class, and were manually cleaned. On purpose, the training images were not cleaned, and thus still contain some amount of noise. This comes mostly in the form of intense colors and sometimes wrong labels. We believe that real-world computer vision algorithms should be able to cope with such weakly labeled data if they are meant to scale well with the number of classes to recognise. All images were rescaled to have a maximum side length of 512 pixels and smaller ones were

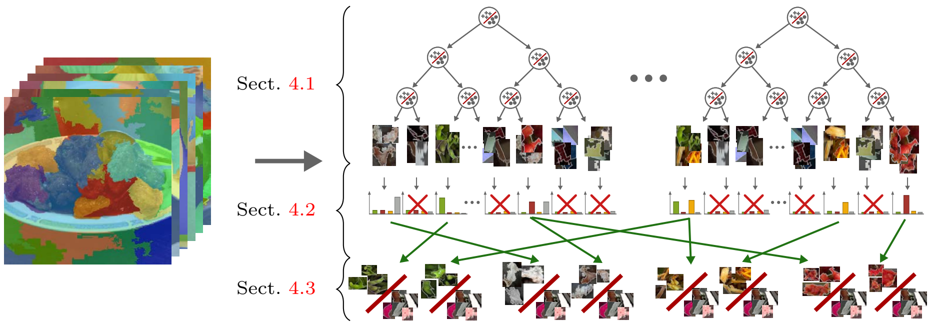


Fig. 3: Overview of our component mining. A Random Forest is used to hierarchically cluster superpixels of the training set. Then, discriminative clusters of superpixels in the leaves are selected and used to train the component models. After mining, the RF is not used anymore.

excluded from the whole process. This leaves us with a dataset of 101’000 real-world images in total, including very diverse but also visually and semantically similar food classes such as *Apple pie*, *Waffles*, *Escargots*, *Sashimi*, *Onion rings*, *Mussels*, *Edamame*, *Paella*, *Risotto*, *Omelette*, *Bibimbap*, *Lobster bisque*, *Eggs benedict*, *Macarons* to name a few. Examples are shown in Fig. 2. The dataset is available for download at `http://www.vision.ee.ethz.ch/datasets/food-101/`.

4 Random Forest Component Mining

In this section we show how we mine discriminative components using Random Forests [14,4] as visualized in Fig. 3. This has two benefits: In contrast to [7,35,38,25,34], components can be mined for all classes jointly because Random Forests are inherently multi-class learners. Compared to [7,8,19] which follow a bottom-up approach and thus need to evaluate all of the several thousands candidate component SVMs to assess how discriminant they are, Random Forest mining instead employs top-down clustering to generate a set of candidate components (Sect. 4.1). Thanks to the class-entropy criterion for choosing split functions, the generation of components is directly related to their discriminative power. We refine the selection of robust discriminative components in a second step (Sect. 4.2) by looking at consistent clusters across the trees of the forest and train robust component models afterwards (Sect. 4.3). The final classification step is then detailed in Sect. 4.4.

4.1 Candidate Component Generation

For generating candidate clusters, we train a weakly supervised Random Forest on superpixels associated with the class label of the image they stem from. By maximizing the information gain in each node, the forest will eventually separate discriminative superpixels from ambiguous ones that occur in several classes. Discriminative superpixels likely end up in the same leaf while non-discriminative ones are scattered.

Let a forest $\mathcal{F} = \{T_t\}$ be a set of trees T_t, each one trained on a random selection of samples (superpixels) $\mathcal{S} = \{s_i = (\mathbf{x}_i, y)\}$ where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector of the sample s_i and y the class label of the corresponding image. For each node n we train a binary decision function $\phi_n : \mathbb{R}^d \rightarrow \{0, 1\}$ that sends each sample to either the left or right sub-tree and splits $\mathcal{S}$ into $\mathcal{S}_l$ and $\mathcal{S}_r$.

While training, at each node, the decision function ϕ is chosen out of a set of randomly generated decision functions $\{\phi_n\}$ so as to maximise the information gain criterion

$$\mathcal{I}(\mathcal{S}, \phi) = H(\mathcal{S}) - \left(\frac{|\mathcal{S}_l|}{|\mathcal{S}|} H(\mathcal{S}_l) + \frac{|\mathcal{S}_r|}{|\mathcal{S}|} H(\mathcal{S}_r) \right), \tag{1}$$

where $H(\cdot)$ is the class entropy of a set of samples. The training continues to split the samples until either a maximum depth is reached, or when too few samples, or samples of a single class are left. In this work we use linear classifiers [3] as decision functions, and more specifically resort to training binary SVMs:

$$\phi(\mathbf{x}) = 1_{[\mathbf{w}^\intercal \mathbf{x} + b > 0]}. \tag{2}$$

We generate different $\phi(\mathbf{x})$ by training them on randomly generated binary class partitions of the class labels in $\mathcal{S}$.

After training the forest, each tree T_t has a set of leaves $L_t = \{l\}$. In the sequel, we denote by $\mathcal{L} = \cup_t L_t$ the set of all leaves in the forest. They constitute the set of candidates for discriminative components. In the next section, we describe how we select the most discriminative ones.

4.2 Mining Components

After training the forest as described in Sect. 4.1, the input space has been partitioned into a set $\mathcal{L}$ of leaves. However, not all leaves have the same discriminative power and several leaves may carry similar information as they were trained independently. In this section, we propose a simple yet effective method to identify a diverse set of discriminative leaves for each class.

Based on the training data, each leaf l is associated with an empirical distribution of class labels $p(y|l)$. Using a validation set, we classify each sample s using the forest, and we define $\delta_{l,s} = 1$ if the sample has reached the leaf l, and 0 otherwise. For each sample, we can easily derive its class confidence score $p(y|s)$ from the statistics of the leaves it reached:

$$p(y|s) = \frac{1}{|\mathcal{F}|} \sum_{l \in \mathcal{L}} \delta_{l,s} \; p(y|l). \tag{3}$$

Note that $\sum_l \delta_{l,s}$ is equal to the number of trees in the forest, *i.e.*, $|\mathcal{F}|$, as a sample reaches a single leaf in each tree.

A high class confidence score implies that most trees were able to separate the sample well from the other classes. To obtain components, we could use these discriminative samples directly in spirit of exemplar SVMs [26]. However, many discriminative samples are very similar. For efficiency, *i.e.*, to reduce the number of component models, it makes sense to identify consistent clusters of discriminative samples instead and train a single model for each cluster.

This is readily possible by exploiting the leaves again. For a single class y, we can evaluate how many discriminative samples are located in each leaf l by considering the following measure:

$$\text{distinctiveness}(l|y) = \sum_s \delta_{l,s}\; p(y|s). \tag{4}$$

Leaves with high distinctiveness are those which collect many discriminative samples (*i.e.*, that have a high class confidence score), thus forming different clusters of discriminative samples. Note that discriminative clusters that are identified by different trees can be easily filtered out by a variation of non-maxima suppression: After sorting the leaves based on their distinctiveness, we ignore models that consist of more than half of the same superpixels as any better scoring leaf. This way, we increase the diversity of components while retaining the strongest ones. Although models with a very similar set of superpixels indicate a very strong component, diversity is more beneficial for classification as this provides richer input to the final classifier.

In Fig. 1 and 7, we show such examples of mined components and study the influence of the number of trees and their depth, but also the number N of discriminative components kept for each food category in Sect. 5.2.

4.3 Training Component Models

For each class, we then select the top N leaves and train for each one a linear binary SVM to act as a *component model*. For training, the most confident samples of class y of a selected leaf act as positive set while a large repository of samples act as negative. To speed-up this process, we perform iterative hard-negative mining. Note that nothing prevents a single leaf to be selected by several classes. This is not a problem at all, since only samples of a single class are used as positives for training a single model.

4.4 Recognition from Mined Components

For classifying an image, we only need to score all of its superpixels using the previously trained component models, instead of applying multi scale sliding window detectors [7,35,38,25,8,19,34,40]. This leaves us with a score vector of $K \times N$ component confidence scores for K classes and N components for each superpixel as illustrated in Fig. 4. In case of a sliding window detector, a standard approach is to max pool scores spatially and then use this representation to train an SVM. We use a spatial pyramid with 3 levels and adopt a slightly different

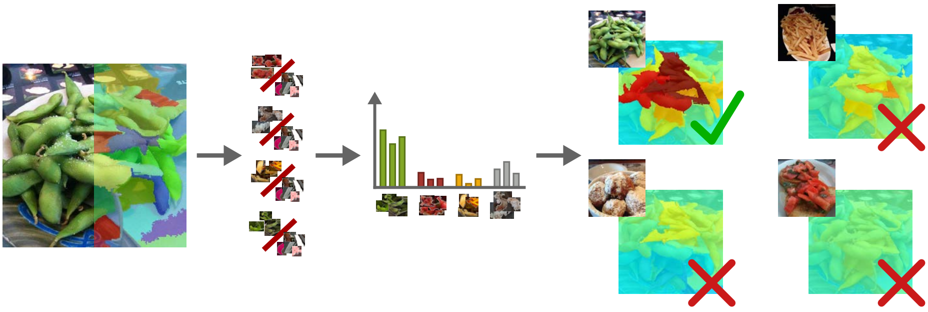


Fig. 4: At classification time, all superpixels of an input image are scored using the component models, afterwards a multi-class SVM with spatial pooling predicts the final class. In this visualisation, we show the confidence scores of edamame, french fries, beignets and bruschetta.

approach for our superpixels: Each superpixel fully contributes to each spatial region it is part of. The scores are then averaged within each region. This loose spatial assignment has proved significantly beneficial for the task of food recognition compared to more elaborate aggregation methods like soft-assignment of superpixels to regions. For final classification, we train a structured-output multi-class SVM using the optimized cutting plane algorithm [17], namely using DLib's [21] implementation.

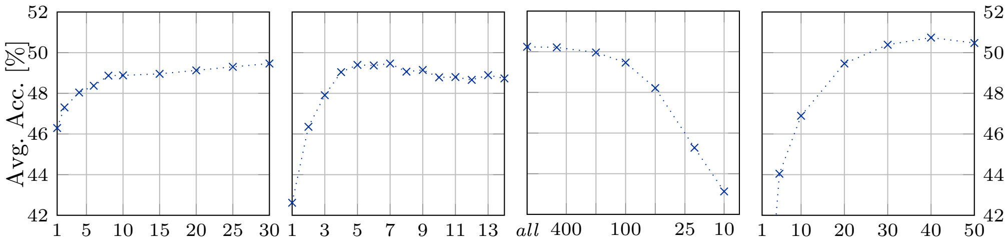
5 Experimental Evaluation

In the following, we refer to our approach as Random Forest Discriminant Components (RFDC) and evaluate it against various methods. For our novel Food-101 dataset (Sect. 3), 750 images of each class are used for training and the remaining 250 for testing. We measure performance with average accuracy, *i.e.* the fraction of test images that are correctly classified. We first give details of our implementation in Sect. 5.1 and analyze then the robustness of our approach with respect to its different parameters in Sect. 5.2. In Sect. 5.3, we compare to baselines and alternative state-of-the-art component-mining algorithms for classification. As our approach is generic and can be directly applied to other classification problems as well, we also evaluate on the MIT-Indoor dataset [31] in Sect. 5.4.

5.1 Implementation Details

We first describe the parameters that we held constant during the evaluation and which had empirically little influence on the overall classification performance.

Superpixels and Features. In this work, we have used the graph-based superpixels of [10]. In practice, setting $\sigma = 0.1$, $k = 300$ and a minimum superpixel size of 1% of the image area yields around 30 superpixels per image, and a total of about



(a) # of Trees (b) Tree depth (c) # of samples per model (d) # of components per class

Fig. 5: Influence of different parameters of RFDC on classification performance on the Food-101 dataset.

2.4 million superpixels in the training set. Changes in those parameters had limited impact on the classification performance. For each superpixel, two feature types are extracted: Dense SURFs [2], which are transformed using signed square-rooting [1], and L*a*b color values. In our experiments it has proved beneficial to also extract features around the superpixels namely within its bounding box, to include more context. Both SURF and color values are encoded using Improved Fisher Vectors [32] as implemented in VlFeat [37] and a GMM with 64 modes. We perform PCA-whitening on both feature channels. In the end the two encoded feature vectors are concatenated, producing a dense vector with 8'576 values.

Component Mining. For component mining, we randomly sample 200'000 superpixels from the 2.4 million to use as a validation set. Each tree is then grown on 200'000 randomly sampled superpixels from the remaining 2.2 million samples. At each node, we sample 100 binary partitions by assigning a random binary label to each present class. For each partition, a binary SVM is learnt, and the SVM that maximizes the criterion in Eq. 1 is kept. The training of SVMs is performed using at most 20'000 superpixels. And the splitting is stopped if a node contains less than 25 samples.

5.2 Influence of Parameters for Component Mining

To measure the influence of the parameters of RFDC, we proceed by fixing the values of the parameters and vary one dimension at the time. By default, we trained 30 trees and mined the parts at depth 7. We then used the top 20 scored component models per class and train each of them using their top 100 most confident samples as positive set.

Forest Parameters. Fig. 5 shows the influence of the number of trees, tree depth, number of samples per model and number of components per class on classification accuracy. RFDC is very robust with respect to those parameters. For instance,

Encoding & Features		**Avg. Acc.** [%]
-	HOG	8.85
BOW	SURF@1024	33.47
BOW	SURF@1024 + Color@256	38.83
IFV	SURF@64	44.79
IFV	Color@64	14.24
IFV	SURF@64 + Color@64	49.40

Table 1: Classification performance for different feature types for RFDC. $@K$ refers to the code book size.

Method	**Avg. Acc.** [%]
Global	
BOW [24]	28.51
IFV [32]	38.88
CNN [23]	56.40
Local	
RF [3]	32.72
RCF [29]	28.46
MLDS (≈ [34])	42.63
RFDC (this paper)	50.76

Table 2: Classification performance measured for the evaluated methods. All component mining approaches use 20 components per class.

increasing the number of trees from 10 to 30 does not make a big difference in accuracy (see Fig. 5a), and tree depth has also little influence beyond 4 levels (Fig. 5b). Using more positive samples to train the component models (Sect. 4.3) improves classification performance of the system, but a plateau is reached beyond 200 samples (Fig. 5c). However, using only 200 positive samples results in significant speed-ups in training. Similar to other approaches [34], Fig. 5d shows that classification performance improves as the number of components per class grows. Also for this parameter the performance saturates. Moreover, the modest improvement in classification accuracy beyond 20 components per class comes with a dramatic increase in feature dimensionality (only worsen by spatial pooling): from 42'420 for 20 components, the dimensionality reaches 106'050 for 50 components and thus heavily impacts memory usage and speed. In conclusion, our RFDC method shows a very strong robustness with respect to its (hyper-) parameters. Fine-tuning of these parameters is therefore not necessary in order to achieve good classification accuracy.

On Features. Using the standard settings as in the previous experiment, we compared different feature types for RFDC. For extracting HOG, we resize the superpixel patches to 64 × 64 pixels. For BOW and IFV encoding, we use the the dictionary sizes as shown in Tab. 1. Unsurprisingly, HOG is not well suited for describing food parts, as their patterns are rather specific. SURFs with BOW encoding yield significant improvement only superseded by IFV encoding.

5.3 Comparison on Food-101

To compare our RFDC approach to different methods, we use 30 trees with a max depth of 5. For mining, we keep 500 positive samples per component and 20 components per class. We compare against the following methods:

Bag-of-Words Histogram (BOW). As a baseline, we follow a classical classification approach using Bag-of-Word histograms of densely-sampled SURF

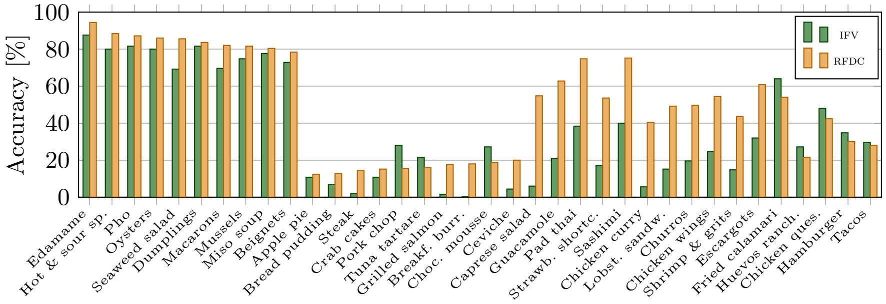


Fig. 6: Selected classification accuracies: The 10 best and 10 worst performing classes are shown as well as 11 classes with the highest improvement and the 8 additional classes for which performance was degraded compared to IFV. The improvements for single classes are visibly more pronounced than degradations.

features, combined with a spatial pyramid [24]. We use 1024 clusters learned with k-means as the visual vocabulary, and 3 levels of spatial pyramid. A structured-output multi-class SVM is then used for classification (see Sect. 4.4).

Improved Fisher Vectors (IFV). To compare against a state-of-the-art classification methods we apply Improved Fisher Vector encoding and spatial pyramids [32] to our problem. For this we employ the same parameters as in [19]. We also use a multi-class SVM for classification.

Random Forest Classification (RF). The Random Forest used for component mining (Sect. 4) can be used directly to predict the food categories, as it is a multi-class classifier. As in [3], we obtain the final classification by aggregating the class confidence score (Eq. 3) of each superpixel s_i and then classify an image $I = \{s_i\}$ using $y^* = \text{argmax}_y \sum_{s_i \in I} p(y|s_i)$. This will highlight the benefit and importance of component mining (Sect. 4.2) and having another SVM for final classification.

Randomized Clustering Forests (RCF). The extremely randomized clustering forest approach of [29] can also be adapted to our problem. The trained RF for component mining can again be used to generate the feature vectors as in [29]. To obtain the final classification, a multi-class SVM is trained on top of these features. This comparison also will show the importance of dedicated component models.

Mid-Level Discriminative Superpixels (MLDS). We implemented the recent approach of [34] for comparison and replaced sliding HOG patches with superpixels. The negative set consists of 500'000 random superpixels and all the superpixels from one class (around 22'500) form the discovery set. We clustered the samples with k-means using a clusters/samples ratio of 1/3. For each class, we discovered discriminative superpixels by letting the algorithm iterate at most 10 times and train each SVM on the top 10 members. For selecting the 20 components per class, we used the discriminativeness measure

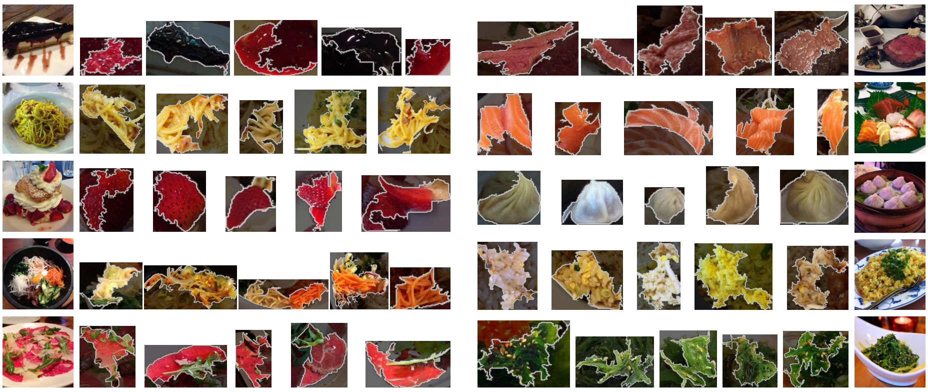


Fig. 7: Examples of discovered components. For each row, an example for the particular dish and examples of discovered components are shown. From top left to bottom right: cheese cake, spaghetti carbonara, strawberry shortcake, bibimbap, beef carpaccio, prime rib, sashimi, dumplings, fried rice and seaweed salad.

as in [34] and Sect. 4.4 for classification. This comparison will demonstrate the benefit of RF component mining.

Convolutional Neural Networks (CNN). We also compare our approach with convolutional neural networks. To this end, we train a deep CNN on our dataset using the architecture of [23] as provided by the Caffe [16] library until it converged (450'000 iterations).

Quantitative Results. We report in Tab. 2 the classification accuracies obtained by the different methods discussed above on the Food-101 dataset. Among global classifiers, IFV significantly outperforms the standard BOW approach by 10%. Switching to local classification is clearly beneficial for the Food-101 dataset. The MLDS approach [34] using strong features on superpixels already gives an improvement of 3.75% with respect to IFV. Looking at the results of Random Forests, we first observe that using them directly for classification performs similar to BOW (about 33% accuracy). The bagging of the random trees is not able to recover from the potentially noisy leaves. Also Randomized Clustering Forests perform at a similar accuracy level. As the number of samples is very limited, the intermediate binary representation is probably too sparse. When using the discriminative component mining together with multi-class SVM classification, we measure an accuracy of 50.76%, an improvement of 8.13% and 11.88% compared to MLDS and IFV, respectively. Also on this dataset, CNN set the state of the art and RFDC are outperformed by a margin of 5.64%. This is paid by a considerably longer training time of six days on a NVIDIA Tesla K20X.

Qualitative Results. In Figs. 1 and 7 we show a few examples of classes and their corresponding mined components. Note how the algorithm is able to find subtle

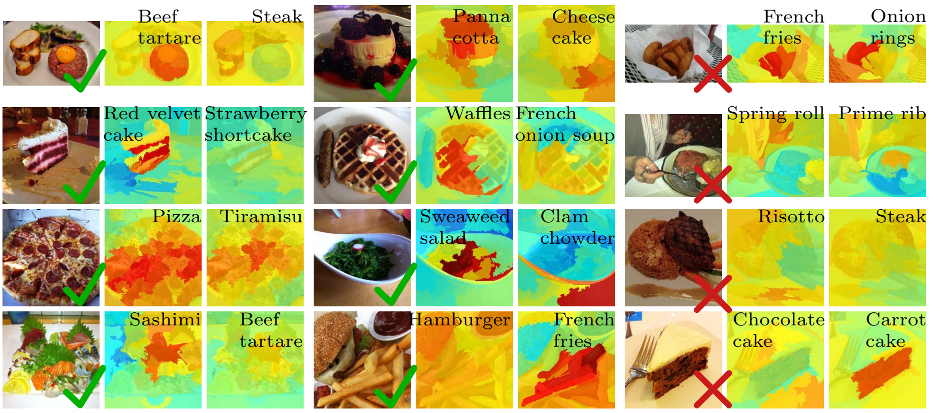


Fig. 8: Examples of the final output of our method. For each correctly classified image, we show the confidence heat map of the true and the second most confident class. For misclassified examples the confidence map of the wrongly predicted class and the true class are shown.

visual components like the fruit compote for the cheese cake, single dumplings, or the strawberries of the strawberry short cake. For other classes, the discriminative visual components show more distinct textures like in the case of spaghetti carbonara, fried rice or meat texture. An interesting visualization is also possible thanks to superpixels. For each class, one can aggregate the component scores and therefore observe which regions in the images are responsible for the final classification. We illustrate such heat maps in Fig. 8. Again, we observe a great correlation between the most confident regions with the actual distinctive elements of each dish. Confusions are often because of visual similarity (onion rings vs. french fries, carrot cake vs. chocolate cake), clutter (prime rib vs. spring roll) or ambiguous examples (steak vs. risotto).

5.4 Results on MIT-Indoor

For running the experiments on the MIT-Indoor dataset, we use the same settings as for Food-101 except, that we sample 100'000 samples per bag. Additionally, we horizontally flip the images in the training set to generate a higher number of samples. For conducting the experiments, we follow the original protocol of [31] with approximately 80 training and 20 testing images per class (*restricted train set*). As this is a rather low number of training examples, we also report the performance on the original test set, but with training on all available training images (*full train set*).

As summarized in Tab. 3, using 50 components per class our method yields 54.40% and 58.36% average accuracy for the restricted and full training set, respectively. While our approach does not match [7] on the restricted train set, the gap gets considerably smaller when training on the full train set. While [7]

Method	**Avg. Acc.** [%]
Part based	
HOG Patches [34]	38.10
BOP [19]	46.10
MI-SVM [25]	46.40
MMDL [38]	50.15
D-Parts [35]	51.40
DMS [7]	64.03
Part based (this paper)	
RFDC (restricted train set)	54.40
RFDC (full training set)	58.36
Global or mixed	
IFV [19]	60.77
IFV + BOP [19]	63.10
IFV + DMS [7]	66.87

Table 3: Recent results of discriminate part mining approaches and global approaches on the MIT-Indoor dataset.

achieves their impressive results with 200 components per class, HOG features and multi scale sliding window detectors, our method evaluates only 50 components on typically 30 superpixels per image. For training, our full pipeline uses around 250 CPU hours (including vocabulary training, i/o *etc.*) with many parallelizable tasks (segmentation, feature extraction and encoding, training of single trees). Approximately 55% of the time is spent on training the forest, 15% for training the component models and 20% for the training of the final classifier.

Compared to other recent approaches, RFDC significantly outperforms [34] as well as all the other very recent sliding window methods of [19,25,38,35]. Note that some of them train their components on external data [25] or have a higher number of components ([35] uses 73 components per class). Clearly, one of the reasons for the achieved performance is the use of stronger features. On the other hand, stronger features can be used here only because our approach needs to evaluate only a small number of superpixels compared to thousands of sliding windows. Still, the full classification time (including feature extraction and Fisher encoding) of one image is around 0.8 seconds using 8 cores, where 70% of the time is spent on encoding and 25% for evaluating the part models.

Interestingly, most previously proposed part-based classification approaches based on sliding windows (or patches) and HOG features typically did not outperform IFV on other datasets until very recently [7]. Our Food-101 dataset (where RFDC outperforms IFV) therefore presents a bias significantly different from available sets, highlighting its interest as a novel benchmark.

6 Conclusion

In this paper, we have introduced a novel large-scale benchmark dataset for recognition of food. We have also presented a novel method based on Random Forests to mine discriminative visual components and efficient classification. We have shown it to outperform state-of-the-art methods on food recognition except for CNN and obtaining competitive results compared to alternative recent part-based classification approaches on the challenging MIT-Indoor dataset.

Acknowledgments We thank Matthias Dantone, Christian Leistner and Jürgen Gall for their helpful comments as well as the anonymous reviewers.

References

1. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR (2012)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. In: ICCV (2006)
3. Bosch, A., Zisserman, A., Munoz, X.: Image Classification using Random Forests and Ferns. In: ICCV (2007)
4. Breiman, L.: Random forests. Machine Learning (2001)
5. Chen, M., Dhingra, K., Wu, W., Yang, L., Sukthankar, R., Yang, J.: PFID: Pittsburgh fast-food image dataset. In: ICIP (2009)
6. Chen, M.y., Yang, Y.h., Ho, C.j., Wang, S.h., Liu, S.m., Chang, E., Yeh, C.h., Ouhyoung, M.: Automatic Chinese food identification and quantity estimation. In: SIGGRAPH Asia 2012 Technical Briefs (2012)
7. Doersch, C., Gupta, A., Efros, A.A.: Mid-level visual element discovery as discriminative mode seeking. In: NIPS (2013)
8. Endres, I., Shih, K., Jiaa, J., Hoiem, D.: Learning Collections of Part Models for Object Recognition. In: CVPR (2013)
9. Felzenszwalb, P.F., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI (2010)
10. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. IJCV (2004)
11. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. PAMI (2011)
12. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
13. Hariharan, B., Malik, J., Ramanan, D.: Discriminative decorrelation for clustering and classification. In: ECCV (2012)
14. Ho, T.K.: Random decision forests. In: ICDAR (1995)
15. Hoashi, H., Joutou, T., Yanai, K.: Image Recognition of 85 Food Categories by Feature Fusion. In: ISM (2010)
16. Jia, Y.: Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/ (2013)
17. Joachims, T., Finley, T., Yu, C.N.J.: Cutting-plane training of structural SVMs. Machine Learning (2009)
18. Joutou, T., Yanai, K.: A food image recognition system with Multiple Kernel Learning. In: ICIP (2009)
19. Juneja, M., Vedaldi, A., Jawahar, C., Zisserman, A.: Blocks That Shout: Distinctive Parts for Scene Classification. In: CVPR (2013)
20. Kawano, Y., Yanai, K.: Real-Time Mobile Food Recognition System. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (2013)
21. King, D.E.: Dlib-ml: A machine learning toolkit. JMLR (2009)
22. Kontschieder, P., Rota Bulò, S., Bischof, H., Pelillo, M.: Structured class-labels in random forests for semantic image labelling. In: ICCV (2011)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
24. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
25. Li, Q., Wu, J., Tu, Z.: Harvesting mid-level visual concepts from large-scale internet images. In: CVPR (2013)

26. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV (2011)
27. Martin, C., Correa, J., Han, H., Allen, H., Rood, J., Champagne, C., Gunturk, B., Bray, G.: Validity of the remote food photography method (RFPM) for estimating energy and nutrient intake in near real-time. Obesity (2011)
28. Matsuda, Y., Hoashi, H., Yanai, K.: Multiple-Food Recognition Considering Co-occurrence Employing Manifold Ranking. In: ICPR (2012)
29. Moosmann, F., Nowak, E., Jurie, F.: Randomized clustering forests for image classification. PAMI (2008)
30. Noronha, J., Hysen, E., Zhang, H., Gajos, K.Z.: Platemate: crowdsourcing nutritional analysis from food photographs. In: ACM Symposium on UI Software and Technology (2011)
31. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: CVPR (2009)
32. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image Classification with the Fisher Vector: Theory and Practice. IJCV (2013)
33. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
34. Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: ECCV (2012)
35. Sun, J., Ponce, J.: Learning discriminative part detectors for image classification and cosegmentation. In: ICCV (2013)
36. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV (2013)
37. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/ (2008)
38. Wang, X., Wang, B., Bai, X., Liu, W., Tu, Z.: Max-margin multiple-instance dictionary learning. In: NIPS (2013)
39. Yang, S.L., Chen, M., Pomerleau, D., Sukthankar, R.: Food recognition using statistics of pairwise local features. In: CVPR (2010)
40. Yao, B., Khosla, A., Fei-Fei, L.: Combining randomization and discrimination for fine-grained image categorization. In: CVPR (2011)