Efficient Mining of Frequent and Distinctive Feature Configurations

Till Quack[1] Vittorio Ferrari[2] Bastian Leibe[1] Luc Van Gool[1,3]

[1]ETH Zurich
Zurich, Switzerland
{tquack,bleibe}@vision.ee.ethz.ch

[2]University of Oxford
Oxford, UK
ferrari@robots.ox.ac.uk

[3]KU Leuven
Leuven, Belgium
vangool@esat.kuleuven.be

Abstract

We present a novel approach to automatically find spatial configurations of local features occurring frequently on instances of a given object class, and rarely on the background. The approach is based on computationally efficient data mining techniques and can find frequent configurations among tens of thousands of candidates within seconds. Based on the mined configurations we develop a method to select features which have high probability of lying on previously unseen instances of the object class. The technique is meant as an intermediate processing layer to filter the large amount of clutter features returned by low-level feature extraction, and hence to facilitate the tasks of higher-level processing stages such as object detection.

1. Introduction

Local features are at the heart of the most successful approaches to object class detection and image classification [2, 6, 7, 9, 11, 17, 21]. After learning a class model from training images, these methods are capable of detecting whether a novel object instance is present in a previously unseen test image [7, 11]. Several recent methods go even a step further by *localizing* novel objects up to a bounding-box [2, 6, 17] or their very outlines [20]. These methods are robust to clutter, scale changes, and missing object parts - properties which stem from the advantageous characteristics of local features. However, these advantages come at a price. The local feature extractor is run beforehand and without prior knowledge of the object class. As a result, on a typical image it returns a large number of features, out of which only some fraction lie the object of interest. Especially when the object appears small in the image, the total set of features has a low signal-to-noise ratio. This imposes a great burden on object detectors and other higher-level processes, as they have to find their way to the object through a sea of background features.

In this paper we propose a novel method to filter this large mass of features. It selects features which have high probability of lying on instances of the object class of interest. Our technique is intended as an intermediate layer between feature extraction and object detection. The filtered set of features our method delivers can then be fed into a higher-level object detector. Thanks to this, it starts from a much higher signal-to-noise ratio, and its performance is likely to improve. We expect our method to lead to lower false-positive rates, and possibly also higher detection rates. Besides, starting from a cleaner set of features is likely to benefit other tasks as well, such as segmenting objects from the background, or determining their pose.

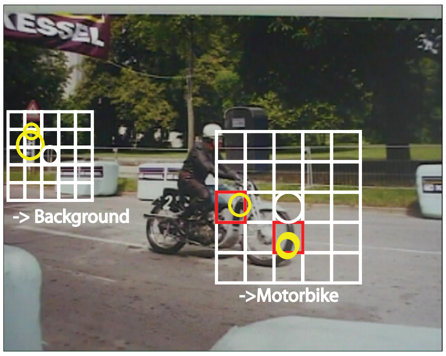

Figure 1. *Example of mined rules: on the left a frequent configuration which infers background, on the right a configuration which infers the object motorbike.*

Our method is based on data mining rather than learning techniques more popular in Computer Vision, such as SVMs. It inputs a set of positive training images, containing different instances of the object class, and a set of negative background images. We organize local features in semi-local neighborhoods and express these in a way suitable for data mining. We adopt a Frequent Itemset Mining algorithm [3], which *efficiently* analyzes the large set of all neighborhoods and returns spatial configurations of local features frequently re-occurring over the training images. From these frequent spatial configurations we collect discriminative Association Rules [3]. These rules infer the presence of the object in positive images with high confidence and fire only rarely on background images. Figure 1 shows two typical feature configurations and the corresponding rules produced by our miner. One rule infers the presence of the motorbike, while the other corresponds to a feature configuration mined from the background. When given a novel image, we first match the mined configurations to it, and then we associate a confidence value to each feature expressing how likely it is to lie on an instance of the object class. This is obtained by accumulating the activation scores of all matched configurations involving the feature.

Our approach has several advantages. First of all, the mining algorithm is designed for scalability and allows to process large training sets rapidly. Moreover, the set of rules collected from the data in this fashion are discriminative and easy to interpret. Indeed, by considering spatial configurations of neighboring features we gain higher discriminative power compared to individual features. A single local feature, even from an informative configuration, might not be distinctive enough and occur frequently also on the background. In addition, the rules often capture configurations of local features corresponding to semantic object parts, such as motorbike wheels (figure 3). The per-feature confidence values produced by our approach effectively prune away the majority of background features, and therefore act as a valuable focus-of-attention mechanism for the benefit of subsequent object detectors, e.g. [2, 11, 17].

Related works. Our work relates to two strands of research: object recognition in computer vision, and data mining.

The idea of using spatial configurations of local features is widely used in object class recognition. The *constellation model* [10] models the spatial arrangement of local features as a joint probability distribution. Inference in this fully connected model has high computational complexity and thus supports only a few features in practice. Fergus *et al.* thus suggest a simplified and more efficient star topology in [11].

Closer to our approach is the work of Lazebnik *et al.*, who propose semi-local arrangements of affine features for object detection [16]. Their method builds directly on features, without vector quantization, and starts by detecting geometrically stable triples of regions in pairs of images. The candidate pairs are summarized by a description which averages over their geometric arrangement. This description is validated on other examples and, if found repeatedly, used for recognition. Our approach instead, builds on vector-quantized features, defines a scale invariant tiled neighborhood, and employs established data mining techniques to find recurring neighborhoods. In addition to being computationally much more efficient, this allows for more variability in the feature appearances. We avoid searching over pairs of images, and mine the whole, large dataset globally *at once*.

The video mining method proposed by Sivic and Zisserman [21] is the most similar work to ours, in that they also build on local neighborhoods of quantized local features. However, the neighborhoods are in their case always of fixed size (*e.g.* the 20 nearest neighbors to a feature). Each neighborhood is expressed as a simple, orderless bag-of-words, represented by a binary indicator vector. Mining proceeds by computing the dot-product between all pairs of neighborhoods and setting a threshold on the resulting number of quantized features they have in common. Our work has several advantages over [21]. First, the neighborhood sizes are based on the scale of the local features, and hence adapt to the image content. Second, by tiling the neighborhood we also include information about feature locations. Third, our mining method avoids the inefficient pairwise matching of neighborhoods over the whole dataset. Fourth, we mine neighborhoods which are distinctive against background images, in addition to occurring frequently over the target objects as those of [21]. Finally, we demonstrate our method on object *classes* rather than specific objects.

The data mining community employed frequent itemset mining and association rules mostly on text data. Only very few approaches have tried to adapt these techniques to visual data. [23] mines databases of annotated images using a diverse set of features such as keywords, file type, and global color and texture features. The focus is on finding hidden correlations between the different modalities of the data, rather than on the visual data itself. In [22] an extended association rule mining algorithm was used to mine spatial associations between texture tile classes in aerial images (e.g. forest, city). In this paper, we bring these promising techniques to the domain of object class detection.

The remainder of this paper is organized as follows. Section 2 describes our approach to mining frequent spatial configurations of local features from training images. In section 3 we determine the confidence that features appearing in new images cover an instance of the object class. An extensive experimental evaluation is reported in section 4, demonstrating our approach primes features lying on class instances and discards background ones.

2. Frequent Feature Configurations

Our technique for mining frequent feature configurations can be summarized as follows. The training set is composed of positive images, containing object instances annotated by a bounding-box, and of negative images, which do not contain any instance of the class of interest. First, a large number of spatial configurations of local image features are collected from all training images. An efficient mining algorithm is then used to select frequently occurring configurations from this large set. The next step transforms these frequent spatial configurations into association rules. These rules are built by selecting frequent spatial configurations which imply the presence of the object class with high confidence, while at the same time are discriminative against clutter (i.e. they occur rarely on the negative images or on non-object areas of the positive images). These *discriminative rules* are the building blocks for a generating class-specific confidence values for features of novel images. These convey the probability that each feature belongs to an instance of the object class (section 3).

The following sections give a detailed description of the individual layers of the our mining system. We start by summarizing the most important concepts and the terminology of association rule mining.

2.1. Frequent Itemsets and Association Rules

Frequent Itemsets. Originally, frequent itemset mining algorithms were developed to solve problems in market basket analysis. The task consists of detecting rules in large amounts (millions) of customer transactions, where the rules describe the probability that a customer buys item(s) B, given that he has already item(s) A in his shopping basket. More precisely, as shown in [3] the problem can be formulated as follows.

Let $I = \{i_1 \dots i_p\}$ be a set of p *items*. We call m*-itemset* a subset A of I with m items. A transaction is an itemset $T \subseteq I$ with a transaction identifier $tid(T)$. A transaction database $D = \{T_1 \dots T_n\}$ is a set of transactions with unique identifiers $tid\,\{T_i\}$. We say that a transaction T *supports* an itemset A, if $A \subseteq T$. We can now define the support of an itemset $A \in D$ in the transaction database D as follows:

$$supp(A) = \frac{|\{T \in D | A \subseteq T\}|}{|D|} \in [0, 1]$$

An itemset A is called *frequent* in D if $supp(A) \geq s_{min}$ where s_{min} is a threshold for the minimal support. Frequent itemsets are subject to the monotonicity property: all m-subsets of frequent $(m+1)$-sets are also frequent. The APriori algorithm [3] takes advantage of the monotonicity property to find frequent itemsets very quickly.

Association rules. An association rule is an expression $A \to B$ where A and B are itemsets (of any length) and $A \cap B = \varnothing$. The quality of a rule can be described in the *support-confidence framework*. The support of a rule

$$supp(A \to B) = supp(A \cup B) = \frac{|\{T \in D | (A \cup B) \subseteq T\}|}{|D|}$$

measures the statistical significance of a rule.

The confidence of a rule

$$conf(A \to B) = \frac{supp(A \cup B)}{supp(A)} = \frac{|\{T \in D | (A \cup B) \subseteq T\}|}{|\{T \in D | A \subseteq T\}|} \tag{1}$$

is a measure of the strength of the implication $A \to B$. The left-hand side of a rule is called *antecedent*, the right-hand side is the *consequent*. Note that the confidence can be seen as a maximum likelihood estimate of the conditional probability that B is true given that A is true [14].

Association rules have several desirable properties. Thanks to the efficient frequent itemset mining method they can be extracted even from very large bodies of data (see section 4). The rule notation is easily interpretable and can be used to gain global insights into large datasets or can be analyzed by experts. These properties have led to their application in several fields such as web usage mining [5] or document analysis [15]. In this paper, we extend the rule-based approach to visual data, and in particular to object detection.

2.2. Local Features and Appearance Codebooks

The lowest layer of our system is built on a set of local features extracted in each image. We use a Difference of Gaussian (DoG) detector to extract regions and the SIFT descriptor [18] to describe their appearance. The SIFT feature vectors are clustered into an appearance codebook (or *visual vocabulary*) with a hierarchical agglomerative clustering method [2]. The use of a codebook representation has recently become very popular [2, 7, 17, 21], since it allows efficient feature matching and captures the variability of a particular feature type (often called *visual word*). Nevertheless, such a description is usually not *semantic*: it does not entail a segmentation of the appearance space into meaningful and distinct object parts (e.g. car wheels, or mug handles).

In order to cope with the inherent uncertainty of the unsupervised clustering process, we *soft-match* each feature by assigning it to all codebook clusters whose center c is closer than a distance threshold d_{min}. This yields a description of each region R_i by a set of codebook labels

$$\zeta_i = \{c_j \mid d(R_i, c_j) < d_{min}\ ,\ j \in 1 \dots N\} \tag{2}$$

where N is the total number of appearance clusters.

2.3. Neighborhood-based Image Description

The second layer of our system builds an image representation from the codebook labels. The simplest representation would be a global histogram, *i.e.* a *bag of features* [7]. However, we aim at unsupervised mining and at learning useful representations for object classes. In this setting, a more informative description is necessary. Encoding not only the presence of visual words, but also their spatial arrangement yields a much stronger descriptor. Thus, we describe each image as a set of semi-local neighborhoods.

Several methods have been proposed to sample spatial neighborhoods from an image. In [6] a sliding-window mechanism samples windows at fixed location and scale steps, followed by a spatial tiling of the windows. The very different approach [21] defines a neighborhood around each region R_c. This is represented as the unordered set of the k nearest regions, without storing any spatial information (*k-neighborhoods*).

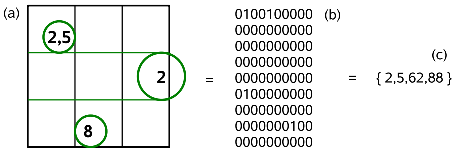

Figure 2. *(a) An example neighborhood with 9 tiles and 10 appearance clusters. Circles represent local features, and numbers indicate the appearance cluster(s) they are assigned to. (b) Activation vector. (c) Transaction.*

Our approach tries to combine the best of both. We rely on the sampling of the feature extractor to define the locations R_c of the neighborhood centers. However, instead of using a k-neighborhood we use the scale of the central region R_c to define the size of the neighborhood. More precisely, all regions falling within a square of side proportional to the scale of R_c are inside the neighborhood. Subsequently, each neighborhood is split into Q tiles as shown in Figure 2a. For each tile we create an activation vector indicating which visual words it contains[1]. The resulting Q activation vectors are concatenated to form the neighborhood descriptor: a $(N * Q)$-dimensional sparse binary vector. Figure 2b shows a neighborhood descriptor for $N = 10$ and $Q = 9$. Note how in this example the top-left region is soft-matched to appearance clusters 2 and 5. The activation vector can equivalently be written as a list of non-zero indices – or, in itemset mining terminology, as a *transaction* (figure 2c). Note how neighborhoods can be made rotation invariant by aligning the tile grid with the dominant orientation of R_c.

Since we form a neighborhood for every region in every training image, this results in a very large number of neighborhoods (or transactions). The training sets in section 4 have between 26000 and 74000 transactions.

2.4. Mining Frequent and Distinct Configurations

Equipped with the tools introduced in the previous sections, we can now find frequent configurations of visual words efficiently. We are especially interested in mining *distinctive* configurations, which appear frequently on the object and rarely on the background.

As discussed above, each neighborhood is described by a list of non-zero indices, and generates a transaction. The input to the mining algorithm (section 2.1) is the database containing all transactions. In order to discriminate against background data, we add transactions from the negative training set to the database. All transactions originating from instances of the object class are assigned the label "object" as an additional item, while we append the item "background' to background transactions. For example, the complete transaction for the neighborhood in figure 2 is $\{2, 5, 62, 88, object\}$ (assuming it lies on an object).

We run the APriori [3] algorithm on the transaction database in order to mine frequent itemsets and association rules. We filter the resulting rules to keep only those which infer the object label with high confidence, *i.e.*

$$conf(\mathcal{C} \rightarrow object) > conf_{min} \qquad (3)$$

where the antecedent $\mathcal{C}$ is a frequent configuration and $conf_{min}$ is a confidence threshold. Notice how a rule does *not* have a high confidence if it appears frequently on both objects and background. This can be understood by inspecting equation (1), where confidence expresses the strength of the implication $\mathcal{C} \rightarrow \mathrm{object}$ (see section 2.1). Hence, our approach finds frequent *and* distinctive feature configurations. Moreover, frequent itemset mining finds these prototypical configurations very efficiently from the immense search space of all 2^{N*Q} possible configurations (typically $N \simeq 3000$ and $Q \simeq 16$; see section 4 for computation times).

As additional advantage, many of the mined rules have semantic qualities, as shown in figure 3. The top left image shows activations of one particular rule on the Caltech-4 set [10] used to mine rules for motorbikes. Activations on two novel test images are shown in the second and third row (see next section for how to match the mined configurations to new images). The regions matching the antecedent $\mathcal{C}$ of the rule are marked in yellow. The central region R_c defining the neighborhood $\mathcal{P}$ is shown in white[2]. Notice the variability in the shape and appearance of the motorbikes, and the different scales of the neighborhoods (automatically adapting to the image data). The rule in the figure is $\{32909, 34622, 46292\} \rightarrow motorbike$ with $s = 3\%$ support and $c = 100\%$ confidence. This rule is one of the most discriminant found for *motorbike*. This makes sense, as wheels are its most characteristic parts. Similar observations can be made for the giraffes in the right column.

3. Determining class-specific feature confidences in novel images

The frequent feature configurations $\mathcal{C}$ mined from the neighborhoods in the training images represent frequent and discriminant fragments of an object class. They describe neighborhoods characteristic for the object class.

Given a new test image, we can now match the mined configurations to it, and hence discover features lying on instances of the object class. To achieve this, we start by generating all neighborhoods $\mathcal{P}$ of the new image (one for each region, as described in section 2.3). Every mined configuration $\mathcal{C}$ is now matched to each image neighborhood $\mathcal{P}$ as follows. A configuration can be written as a sparse activation vector. Hence, the test image neighborhoods can be matched efficiently by a sparse dot-product:

$$m(\mathcal{C}, \mathcal{P}) = \begin{cases} 1 & \text{if } \mathcal{C} * \mathcal{P} = |\mathcal{C}| \\ 0 & \text{if } \mathcal{C} * \mathcal{P} \neq |\mathcal{C}| \end{cases} \qquad (4)$$

[1] We do not count multiple occurrences of the same visual word in a particular tile, *i.e.* we work with sets instead of bags.

[2] R_c is not part of the rule. In this example the rule consists of the yellow regions only.

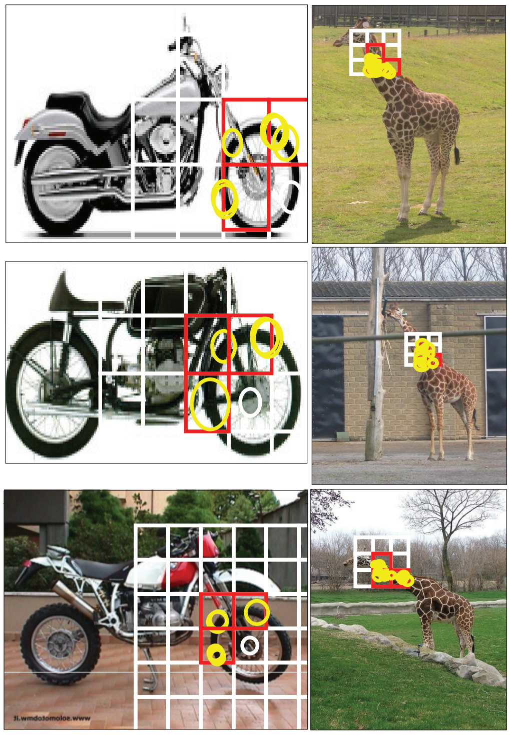

Figure 3. *Discriminant Frequent Spatial Configurations. First row: examples from training set. Second/third row: examples of activations on the test-set. Note: R_c (white) is not part of the rule.*

where $|\mathcal{C}|$ is the number of features in $\mathcal{C}$, and $m(\mathcal{C}, \mathcal{P}) = 1$ indicates a match. In other words, a frequent configuration $\mathcal{C}$ matches a candidate neighborhood $\mathcal{P}$ if their dot product equals the number of visual words in $\mathcal{C}$.

From matched neighborhoods of the test image we can derive a measure of the probability for a feature to lie on an instance of the object class. This measure effectively enables to pre-select features lying on the object, and hence it can substantially ease the life of a subsequent object detector. Thanks to this, the latter can focus on higher level tasks, such as localizing the object up to a bounding-box, determining its precise extent (outlines), its pose, a part decomposition, and so on. We compute this class-specific feature confidence measure as follows. For each feature in the image, we count how often it is part of a matched neighborhood. The more matched configurations a features participates into, the more it is likely to cover part of an object instance. More precisely, the confidence measure for each feature R_i is defined as:

$$conf(R_i) = \frac{1}{M * V} \sum_{\mathcal{C}} \sum_{\{\mathcal{P} | R_i \in \mathcal{P}\}} \frac{1}{k} * m(\mathcal{C}, \mathcal{P}) \qquad (5)$$

where M is the number of configurations mined on the training data, V is the number of neighborhoods in the test image, k is the number of appearance clusters to which R_i was soft-assigned (equation (2)).

4. Results

We present results on four diverse object classes. After discussing the quality of the results via some visual examples, we perform a quantitative performance evaluation. The experiments are conducted on the following datasets. The objects in the positive training images were annotated by a bounding-box, except for the *TUD Motorbikes* where full images without bounding box were used for training.

ETHZ Giraffes. Training was conducted on 93 images of giraffes we downloaded from Google Images. No background training data was used in this case. The positive test images are the 87 Giraffes from the ETHZ Shape Classes dataset [12]. All 168 images of the other classes from [12] are used as negative test set (as done for object detection from hand-drawings by [12]).

GRAZ Bikes. All training data and the positive test set are as defined in the paper which originally proposed this dataset [19]. As negative test set we took the first 200 images from the CALTECH-101 background [8] class. This negative test set is used as well with all following datasets.

TUD Motorbikes. The TUD Motorbikes dataset [1] consists of 115 images containing 125 motorbikes, which we used as positive test set. The positive training images are the Caltech-4 motorbikes [10] (no bounding-boxes given). As background training set we randomly picked 200 images from the CALTECH-256 [13] background class.

CALTECH Cars Rear. This dataset features 126 rear-views of cars and 1155 street scenes without cars, used as training set. Moreover, the dataset also provides a test set of 526 images containing cars, as described in [10].

The first three datasets are particularly challenging, as objects appear in severely cluttered images, and present scale and intra-class variations. Moreover, the GRAZ Bikes and TUD Motorbikes are partially occluded in several images. The CALTECH Cars are somewhat easier, in that they appear rather centered in the images and vary only moderately in scale.

4.1. Visual Examples

We present here visual examples to demonstrate the quality of the mined feature configurations (section 2), and of features selected based on the confidence values our approach delivers (section 3). Figure 4 shows several test images, with all overlaid features having a confidence (equation 5) above 20% of the maximum possible value. These features belong to configurations deemed frequent and discriminative by our method. The brighter the color of a feature, the higher its confidence.

The large majority of features are systematically selected on the object, in spite of scale changes, clutter, and intra-class variations. It is particularly interesting to notice how the selected features adapt to the class so as to cover its most discriminative parts. For bikes, the rather structural configurations of frame parts and wheel fragments dominate, whereas for giraffes the pattern of the fur is selected (*i.e.* the miner adapts to behave like a texture detector). Besides, notice how our measure effectively selects object features, and discards background ones. These results confirm that our approach effectively primes object features while pruning away the majority of background ones. Hence, it is a valuable intermediate step before applying higher-level processing such as object localization algorithms. This is particularly interesting for the motorbikes set, where we trained without bounding boxes directly from the CALTECH images. This shows that we can mine relevant rules without bounding boxes, when the training objects are rather centered and there is limited background clutter.

4.2. Quantitative Evaluation of Feature Selection

We quantify the performance of our method for assigning class-specific confidences to features, based on two experiments. In the first experiment we measure *bounding box hit rate* (BBHR) over the positive test sets. A bounding-box hit is counted if more than k features selected by our method lie on the object (inside the bounding box). Hence, BBHR is the number of BBH divided by the total number of object instances in the positive test set. To perform this evaluation we use ground-truth bounding-box annotations available for the test images (these were *not* used to produce the results). The rationale behind the BBHR measure is that the later processes our method is intended to aid, need at least a certain number of features to operate reliably (e.g recognition - deciding whether the object is actually present in the image, or localization - determining a bounding-box framing the object). We set BBHR in relation with the false positive rate (FPR). This is the number of selected features lying outside the bounding box, divided by the total number of selected features in the image (averaged over all positive test images). Essentially FPR measures the (inverse) signal-to-noise ratio output by our method, *i.e.* the proportion of irrelevant features it delivers (the lower the better). We compare our method against a baseline, where the confidence for a feature is computed as follows. For each visual word in the codebook we count how many times it occurs inside the bounding-box annotations of the training data. This way a visual word, which appears often on the training objects is weighted higher. On a test image, we match features to the codebook and define BBHR by summing up the weighted matches for each feature. That is, instead of using configurations of features like our system does, the baseline consists of weighted single feature matches – essentially a bag-of-words scheme. This allows to compare our method to the default input to an object recognition system.

Figure 5 shows FPR on the y-axis and BBHR on the x-axis, for $k = 5$ and for each dataset. The error bars show the standard deviation of the FPR at a given BBHR. Curves are generated by varying the selection threshold over the feature confidences. As the plots show, our feature selection method is very precise, in that it consistently delivers a low FPR (always below 20%, but for high BBHR on the Cars Rear dataset, where it grows to a moderate 35%). This is an important characteristic, because it enables later processes to rely on a clean input, composed of a large majority of features on the object. This appears especially valuable when compared to the low signal-to-noise ratio of the initially extracted features (there are typically $500 - 1000$ features in an image, out of which about $10 - 200$ lie on the object). The experiments also reveal the substantial performance improvement over the baseline, which we outperform substantially.

The feature selection ability comes at a low price in terms of missed objects: on three of the datasets our method selected at least 5 features (typically many more, as in figure 4) on about 90% of the object instances. The lower BBHR on the TUD Motorbikes might be due to an excessively high support threshold for mining or a bad visual vocabulary, and is be the subject of further investigation.

The second experiment evaluates our method on the negative test sets (*i.e.* on image without any instance of the object class). The idea is to measure how distinctive the method is: does it select very few features on negative images? This is relevant because the number of features selected on negative images relates to the computational resources the later processing stages will waste on irrelevant data (and to the chances they will get confused and produce wrong results). Figure 6 reports the percentage of negative images (y-axis) where at most v features are selected (x-axis). The feature selection threshold is left fixed for each curve, to the one yielding $70\%/90\%$ BBHR on the positive dataset (a sensible operating point). As the plots show, at 70% BBHR the method returns extremely few features on the negative images of giraffes and bikes (on 90% of the images it returns less than 3 features). As in the previous experiment, the performance is lower on Motorbikes, but it remains good (in 70% of the images it returns less than 8 features). As expected, at the challenging operating point of 90% BBHR the method returns more features. Nevertheless, it remains distinctive even in this case: 1 in 3 negative images have no selected features, and 70% of the images have less than 10 (remember, we start from $500 - 1000$). The baseline is evaluated in the same manner as for the BBHR plots, and it performs considerably worse than our method.

4.3. Computation times

The CPU-time measurements are given in table 1. The time is measured for the frequent itemset mining stage including rule creation, but after feature extraction and neighborhoods construction. This because the required process-

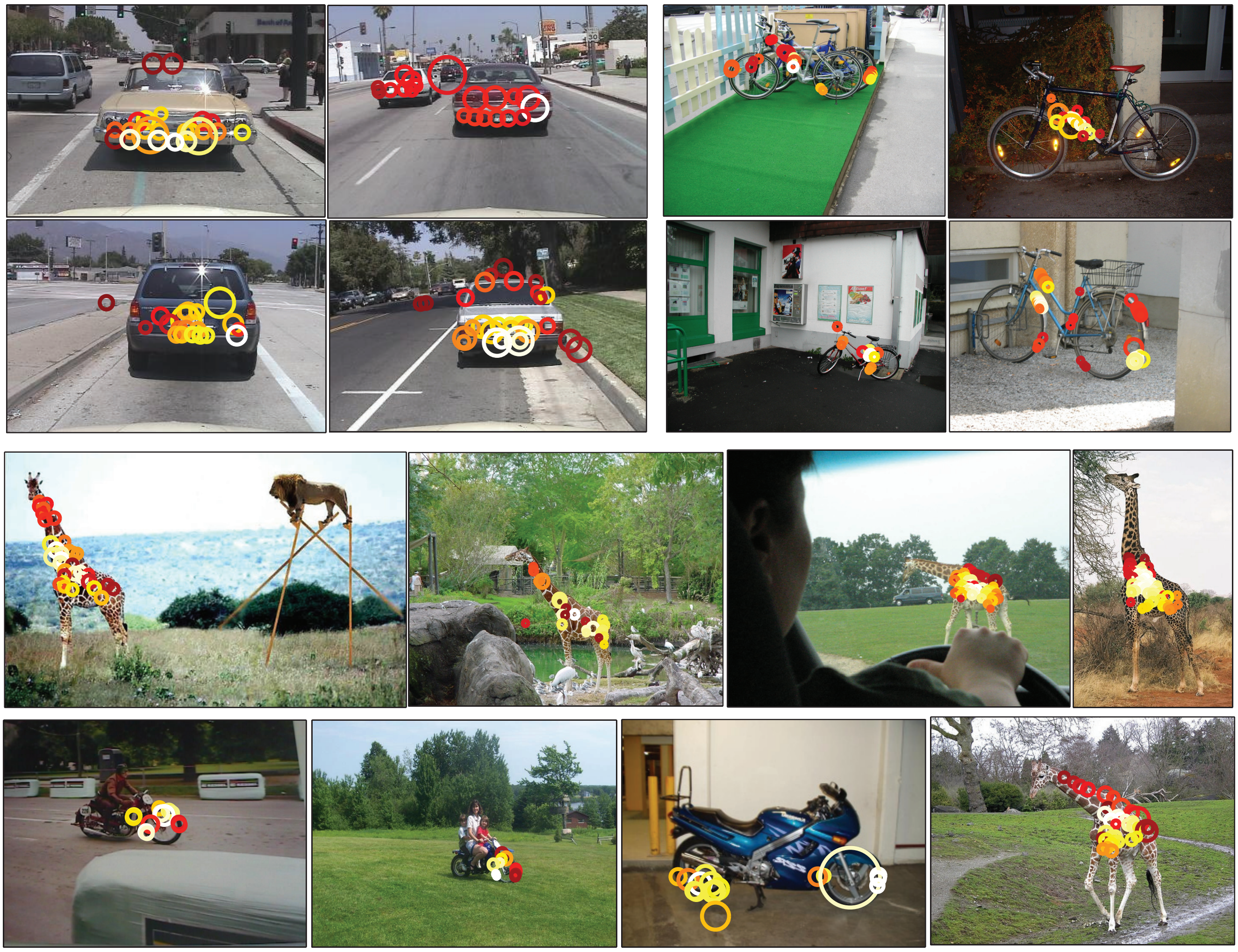

Figure 4. *Results: Visual Examples. (See text for discussion.)*

Data	T	$supp_{min}/conf_{min}$	Q	t CPU
Giraffes	26054	0.20% / 100%	9	2.58 s
Bikes	42390	0.25% / 95%	9	0.91 s
Motorbikes	29001	0.28% / 100%	9	0.90 s
Cars Rear	74296	0.1% / 90%	9	53.02 s

Table 1. *Statistics for the mining experiments. Columns: Number of Transactions T, minimal support and confidence thresholds, number of tiles Q, CPU time (in seconds).*

ing can be done offline and the required time scales linearly with the number of images. For the mining we use an implementation of the APriori algorithm from [4]. All experiments were done on a modern PC. These measurements demonstrate the scalability of our mining approach, where the most characteristic feature configurations can be extracted from tens of thousands of candidates in a matter of seconds. The mined configurations might be used readily within other frameworks. Table 1 also summarizes the mining parameters used for each dataset.

Conclusions We have presented an efficient data mining approach to detect frequent and distinctive feature configurations, representative for an object class. Moreover, we have shown how to exploit the mined configurations to measure how likely it is for features of novel test images to lie on an instance of the object class. Through experimental evaluation we have demonstrated that this class-specific confidence measure acts as a good feature selector. Hence, our technique offers a valuable intermediate layer between feature extraction and object detection or other higher-level processes. Future work includes evaluation on larger datasets and the extension of the rule mining approach to less-supervised scenarios (e.g. training images without bounding-box annotation).

Acknowledgments We acknowledge support from EU project CLASS, IST 027978 and Swiss NSF project IM2.

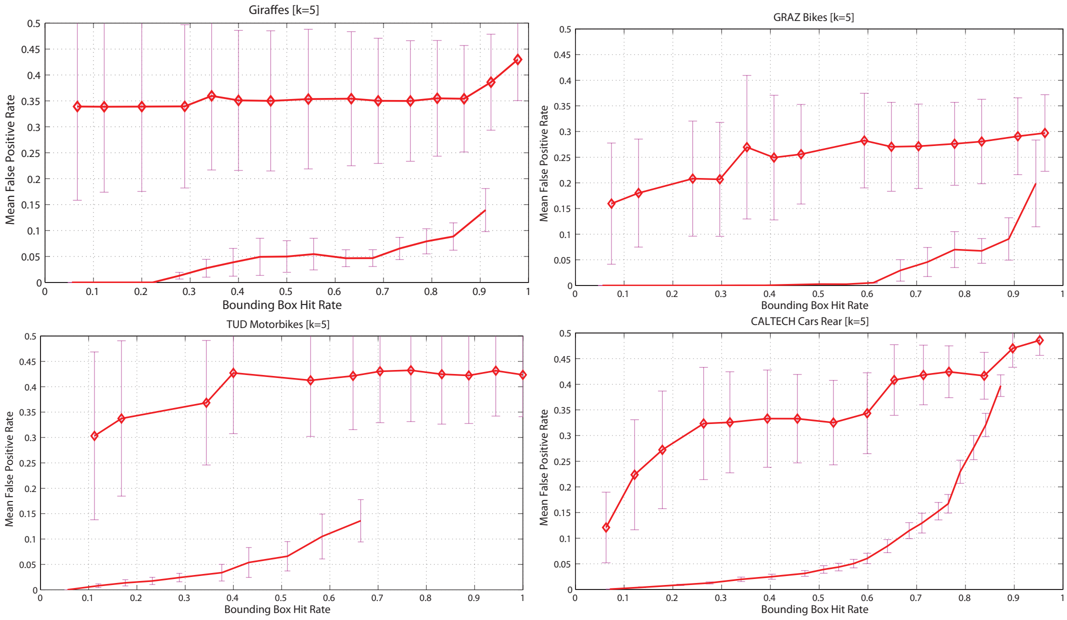

Figure 5. *Bounding box hit rates for Giraffes, Bikes, Motorbikes, and Cars Rear Views (lower is better, baseline with diamond marker).*

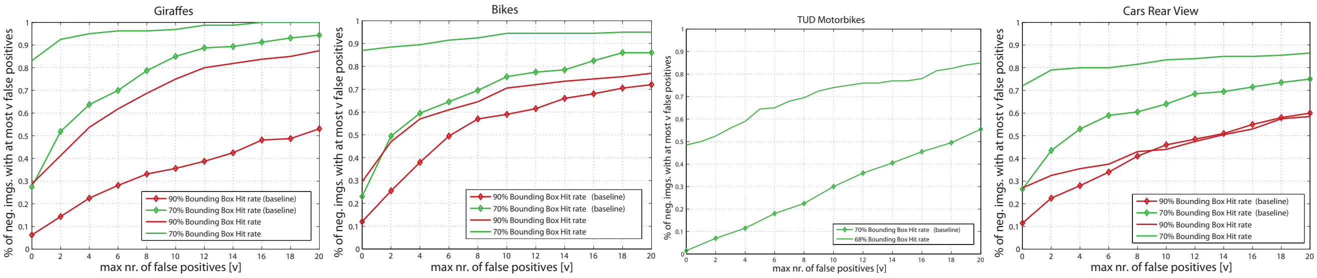

Figure 6. *False positives on negative test images for Giraffes, Bikes, Motorbikes, Cars Rear View (higher is better). For the motorbikes we show the experiment for the threshold at 68% BBHR since this is the maximum we reached.*

References

[1] The pascal object recognition database collection (2005). www.pascal-network.org/challenges/VOC.

[2] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. In *Trans. PAMI*, 2004.

[3] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD'93*.

[4] C. Borgelt. Efficient implementations of apriori and eclat. In *FIMI'03*.

[5] R. Cooley, J. Srivastava, and B. Mobasher. Web mining: Information and pattern discovery on the world wide web. In *ICTAI'93*.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR'05*.

[7] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV SLCV'04*.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an approach tested on 101 object categories. In *CVPR WGMBV'04*.

[9] Feltzenswalb and D. Hutenlocher. Pictorial structures for object recognition. In *IJCV*, 2005.

[10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR'03*.

[11] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR'05*.

[12] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. In *ECCV'06*.

[13] G. Griffin, A. Holub, and P. Perona. The caltech 256. Caltech Technical Report, 2007.

[14] D. Hand. *Principles of Data Mining*. MIT Press, 2001.

[15] J. D. Holt and S. M. Chun. Efficient mining of association rules in text databases. In *ACM CIKM'99*.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *BMVC'04*.

[17] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR'05*.

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2003.

[19] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Generic object recognition with boosting. In *Trans. PAMI*, 2003.

[20] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV'06*.

[21] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *CVPR'04*.

[22] J. Tesic, S. Newsam, and B. S. Manjunath. Mining image datasets using perceptual association rules. In *SIAM'03 Workshop on Mining Scientific and Engineering Datasets*.

[23] O. R. Zaiane, J. Han, Z.-N. Li, and J. Hou. Mining multimedia data. In *CASCON'98*.