Unsupervised Object Discovery via Self-Organisation

Teemu Kinnunen[a], Joni-Kristian Kamarainen[c,*], Lasse Lensu[b], Heikki Kälviäinen[b]

[a]*Department of Media Technology, Aalto University, Finland (http://media.tkk.fi/)*
[b]*Machine Vision and Pattern Recognition Laboratory (MVPR), Department of Information Technology, Lappeenranta University of Technology (LUT) (http://www2.it.lut.fi/mvpr/)*
[c]*LUT Kouvola Unit, Kouvola, Finland*

Abstract

Object discovery in visual object categorisation (VOC) is the problem of automatically assigning class labels to objects appearing in given images. To achieve state-of-the-art results in this task, a large set of positive and negative training images from publicly available benchmark data sets have been used to train discriminative classification methods. The immediate drawback of these methods is the requirement of a vast amount of labelled data. Therefore, the ultimate challenge for visual object categorisation has been recently exposed: unsupervised object discovery, also called unsupervised VOC (UVOC), where the selection of the number of classes and the assignments of given images to these classes are performed automatically. The problem is very challenging and hitherto only a few methods have been proposed. These methods are based on the popular bag-of-features approach and clustering to automatically form the classes. In this paper, we adopt the self-organising principle and replace clustering with the self-organising map (SOM) algorithm. Our method provides results comparable to the state of the art and its advantages, such as non-sensitivity against codebook histogram normalisation, advocate its usage in unsupervised object discovery.

Keywords: Unsupervised object discovery, visual object categorisation, object class detection, self-organising map, bag-of-features

*Corresponding author. Tel.: +358 40 5794605
Email addresses: Teemu.Kinnunen@aalto.fi (Teemu Kinnunen), joni.kamarainen@lut.fi (Joni-Kristian Kamarainen), lasse.lensu@lut.fi (Lasse Lensu), heikki.kalviainen@lut.fi (Heikki Kälviäinen)

1. Introduction

Over the past decade, the category-level object recognition problem has drawn considerable attention within the computer vision research community. As a result, there exist various approaches (e.g. Csurka et al., 2004; Holub et al., 2005; Bar-Hillel and Weinshall, 2008, etc.), and publicly available benchmark data sets such as Caltech-101 (Fei-Fei et al., 2006), Caltech-256 (Griffin et al., 2007) and LabelMe (Russell et al., 2008), and an annual contest: the PASCAL Visual Object Classes Challenge (Everingham et al., 2008, 2009, see, e.g.,). From 2005 to 2010, the classification accuracy of the 101 classes in Caltech-101 increased from 43% (Holub et al., 2005) to 81% (Li et al., 2010), which can be considered as sufficient for certain applications. The best methods are based on discriminative machine learning and require a large amount of training data that need to be labelled and often also annotated by bounding boxes, landmarks, or object boundaries. The laborious manual annotation restricts the research progress, and in response to this, the ultimate challenge of visual object categorisation (VOC) has been exposed: unsupervised visual object categorisation (UVOC) in which the purpose is to automatically find the number of categories in an unlabelled image set and assign images to these categories (Weber et al., 2000).

The best performing VOC methods cannot be used for an unsupervised setting. In their recent work, Tuytelaars et al. (2010) showed that simple baseline methods perform the best. These baseline methods utilise the popular bag-of-features (BoF) approach (see e.g., Csurka et al., 2004), where a codebook of local features is generated from extracted interest point regions, and the images are represented as histograms of these codebook codes. The codebook is generated by clustering the extracted local features, and the classes are estimated by clustering the code histograms to form "a category book". The results of this approach for the unsupervised setting are, however, much worse than for the supervised VOC problem.

In this work, we study self-organisation as a novel solution to unsupervised object discovery. In a neural interpretation of this approach, "neurons" compete for inputs (exhibition), and a single neuron maximises its winning probability by specialising in a specific set of inputs. When this rule is applied as a learning principle, the neuron

best matching to a given input is adapted towards the input and the same adaptation is applied to its neighbours. This leads to self-organisation of the neural lattice. Our algorithm of choice is the self-organising map (SOM) (Kohonen, 1990), also referred to as the Kohonen map. As the computational paradigm, we adopt the bag-of-features approach used by Tuytelaars et al. (2010). The main contributions of this study are as follows: (a) a SOM-based bag-of-features method for the problem of unsupervised object discovery, (b) study of which local feature detectors and descriptors perform the best in this setting, and (c) experimental analysis of our method including a comparison to the state of the art. Specifically, the proposed method achieves accuracy similar to the best method and has some beneficial properties which advocate its use in further research on unsupervised object discovery. As an example, the self-organising map is less sensitive to the success of data normalisation than the k-means algorithm.

1.1. Related work

Object detection from digital images is one of the first research questions investigated in computer vision. However, the problem of visual object categorisation (VOC) in its current form and extent can be tracked back to Burl et al. (1996). Their work has led to numerous methods with various levels of supervision, performance measures for evaluating the VOC methods, and publicly available benchmark data sets. The best published methods achieve the detection accuracy of above 80% for the standard benchmarks containing tens of categories (Fei-Fei et al., 2006; Griffin et al., 2007). The current state of the art can be seen in the results of the annual Pascal VOC Challenge (Everingham et al., 2008, 2009, 2010).

Unsupervised visual object categorisation (UVOC), on the other hand, is a relatively new research topic. For UVOC, the best VOC methods are unfeasible since they are based on discriminative pattern recognition, such as support vector machine classifiers, and need labelled training and validation sets. Tuytelaars et al. (2010) list a few existing methods which are based on the bag-of-features (BoF) principle and omit the spatial information of local features. Recently, the first attempts to use also the spatial information have been reported by Bart et al. (2008) and Sivic et al. (2008), but the results have been reported only for a few classes. In the experiments with the large

databases (Caltech-101 and Caltech-256), the most straightforward approach, in which both the codebook and the category discovery are determined by k-means, reaches superior results as compared to more sophisticated methods (Tuytelaars et al., 2010).

The self-organisation principle was introduced by Kohonen (1990). The Self-Organisation Map (SOM) was originally designed for data visualisation, but it has been successfully applied, for example, to the automatic organisation of Internet news items (Kohonen et al., 1996), patent descriptions (Kohonen et al., 2000), and also images (Laaksonen et al., 2000). The PicSOM by Laaksonen et al. (2000) uses SOM for unsupervised categorisation, but does not otherwise utilise the BoF processing stages.

2. Unsupervised object discovery

A method for unsupervised visual object categorisation (UVOC) should assign the same label to unlabelled images with similar content such as faces, cars, motorbikes, etc. The problem is two-fold: these different semantic classes should be discovered, and the images should be assigned to one class or more. It is noteworthy that humans distinguish 30,000 categories and assign images to these categories with ease and great accuracy (Biederman, 1987). From the pattern recognition point of view, the problem is very difficult: How to extract and represent the important visual content of images in a compact and representative way to make the process computationally feasible for a large number of images and robust for geometric transformations and common image distortions? In addition, how to define and detect the co-occurrence of important visual "patterns" which form a model of a single category and distinguish it from the other categories? Even the evaluation of UVOC methods is difficult and there is no single generally acceptable method. In this section, we first survey the existing performance measures for evaluating UVOC algorithms (Sec. 2.1), and then describe the standard BoF approach (Sec. 2.2).

2.1. Performance evaluation

An image set with the ground truth (correct labels) is needed for evaluating method performance. For this purpose, the standard benchmark data sets can be used, but

problems still remain. For example, how to compare classification results with a different number of discovered classes? These issues have been discussed in the works by Tuytelaars et al. (2010) and Sivic et al. (2008). They applied measures used to evaluate and compare clustering methods. In the following, we review these two works and, in addition, propose a new evaluation method.

Sivic et al. (2008) proposed a performance evaluation method which takes a "categorisation tree" representing the class hierarchy as the input, and computes its performance to represent the true hierarchy. The evaluation protocol utilises the concept of hierarchy, i.e., the categories near the root are more mixed as compared to the leaf nodes, which should represent the pure categories. The performance of a single node, $p(t, i)$, is computed as

$$p(t,i) = \max_i \frac{|GT_i \cap P_t|}{|GT_i \cup P_t|} \ , \tag{1}$$

where GT_i are the ground truth images of the category i, P_t are the images assigned to node t, and $|\cdot|$ denotes the number of images. The average performance, $perf$, is computed as

$$perf = \frac{1}{N_c} \sum_{i=1}^{N_c} \max_t p(t,i) \ , \tag{2}$$

where N_c is the number of categories. The method ultimately chooses nodes that give the best categorisation performance per each object category, and then computes the average over these nodes. The main drawback of this method is that it actually measures the hierarchical decomposition rather than the categorisation performance. It is not clear whether the hierarchy decomposition is relevant for the categorisation task, or whether it is a problem of its own. For example, if the objects of the same category are separated at the upper levels in a tree, it is heavily penalised even though they would end up as two pure leaf nodes.

Tuytelaars et al. (2010) adopted their evaluation strategies from the clustering literature, i.e. how well the produced clusters can map the data to their true labels. They identified two possible cases in the method evaluation: 1) the number of categories is enforced to correspond to the number of ground truth categories or 2) the number of produced categories does not correspond to the number of categories in the original data. For the first case, two simple measures can be used. The first one, "purity", is

computed as

$$purity(X \mid Y) = \sum_{y \in Y} p(y) \max_{x \in X} p(x \mid y) \ , \tag{3}$$

where X stands for the ground truth category labels and Y stands for the cluster labels. In practise, $p(x \mid y)$ is computed empirically from the ground truth label frequencies in each computed category. Purity measures how well a method separates images of one category from all other categories. The second measure is the mutual information (information gain) which is popular in decision tree learning algorithms,

$$I(X \mid Y) = H(X) - H(X \mid Y) \ , \tag{4}$$

which is based on the original entropy of an image set $H(X)$ and the entropy after the categorisation, the conditional entropy $H(X \mid Y)$. The conditional entropy is computed as

$$H(X \mid Y) = \sum_{y \in Y} p(y) \sum_{x \in X} p(x \mid y) \, log \frac{1}{p(x \mid y)} \ , \tag{5}$$

where Y stands for the cluster labels and X for the ground truth labels. Conditional entropy measures how certain one can be that the image actually belongs to the cluster. However, since the term $H(X)$ is constant in (4), the conditional entropy in (5) can be directly used. When the number of clusters increases considerably, the conditional entropy and purity give ideal results. The main problem of these measures is that they can be used for the method comparison only when all methods return the same number of categories. Moreover, the values depend on the total number of images. The main drawback, however, is the limitation that neither of the methods estimate the categorisation accuracy well if the estimated number of categories is not the same as in the ground truth and especially if $|Y| > |X|$. In the ultimate case, every image is its own category, and this produces the perfect performance values $purity = 1$ and $H(Y \mid X) = 0$. Tuytelaars et al. circumvented this undesirable property by introducing an "oracle", which means that there are separate training and test sets to discover the classes. In this case, on the other hand, the normal classification accuracy can be used, as well, and its performance value, the classification accuracy, is more intuitive than the conditional entropy.

Since the performance measures proposed by Sivic et al. (2008) and Tuytelaars et al. (2010) both have some undesirable properties, we also define our own measure. In our case, we directly compute the classification accuracy of the test set. Basically, this can be done directly based on the training data, which measures only the quality of features, or by using the category labels similar to the oracle method by Tuytelaars et al. Our measure will be used in parallel with their method in the experimental section to demonstrate the equivalence. The main advantage of our method is that the classification accuracy is more intuitive than the conditional entropy.

2.2. Bag of features approach

For our method, we adopt the BoF approach used by Tuytelaars et al. (2010) since it is the most prominent approach in supervised VOC (Everingham et al., 2010) and also performs very well in the unsupervised setting. BoF is illustrated in Fig. 1a.

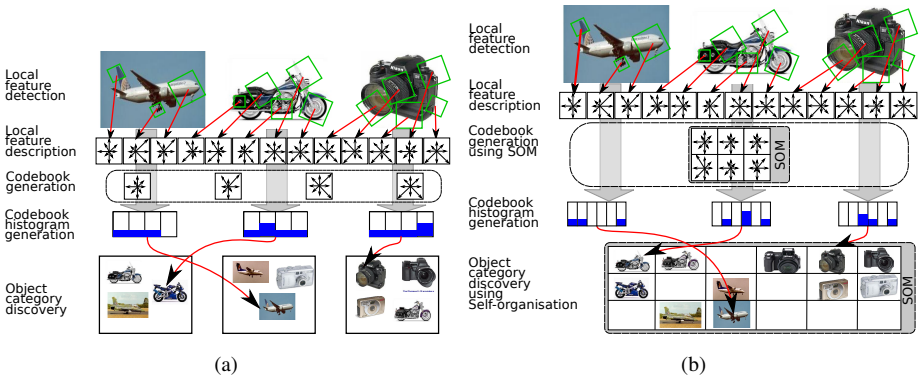


Figure 1: The information flow in (a) the general bag-of-features approach and (b) the self-organisation model for unsupervised object class discovery.

In the first step of BoF, local features are detected using an automatic local feature detector. Next, the local features – regions – are described using a local feature descriptor. The combination of the local feature detection and description is commonly referred to as local feature extraction. This topic is discussed in more detail in Section 3.1. In the third step, a visual codebook is formed by clustering the local feature

descriptors. Next, a histogram of codebook entries is generated for each image by matching the local feature descriptors with the visual codebook and by counting the matches. In the last step, the object classes are automatically discovered using e.g. k-means for the histograms, and images are assigned to these classes.

3. Self-organisation approach

Our method corresponds to the general BoF UVOC structure illustrated in Fig. 1a. The main differences are that we perform the steps for codebook generation and object class discovery by using the SOM algorithm. The refined model is shown in Fig. 1b. In this section, we briefly revisit each step to clarify their role in the approach. Experimental results for the first step, local feature detection and description, are included in this section while the other experimental results are in Sec. 4.

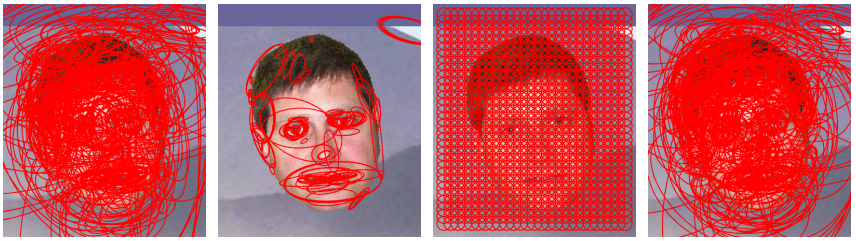
3.1. Local feature detectors and descriptors

A number of local feature detectors and descriptors have been proposed in the literature. A survey and comparison of different detectors can be found in the work of Mikolajczyk et al. (2005b) and for the descriptors in that of Mikolajczyk and Schmid (2005). The comparisons, however, are based on the repeatability and matching performances over different views of the same scenes. Therefore, their applicability to VOC and UVOC is unclear. More explicit VOC evaluations have been carried out by Zhang et al. (2006) and Mikolajczyk et al. (2005a). Their main conclusions were that detector combinations performed better than any single detector, and that the extended SIFT descriptor, the Gradient Location and Orientation Histogram (GLOH) (Mikolajczyk and Schmid, 2005), is slightly superior to others. The better performance using the combinations can also be explained by the increased number of detected features. The drawback of GLOH is that it requires training data to estimate eigenvectors for the required PCA dimensionality reduction step – proper selection of the PCA data can also explain the slightly better performance. Based on the above works, SIFT (difference of Gaussians) can be safely used as the descriptor, but it is justified to investigate which detector is the most suitable for UVOC.

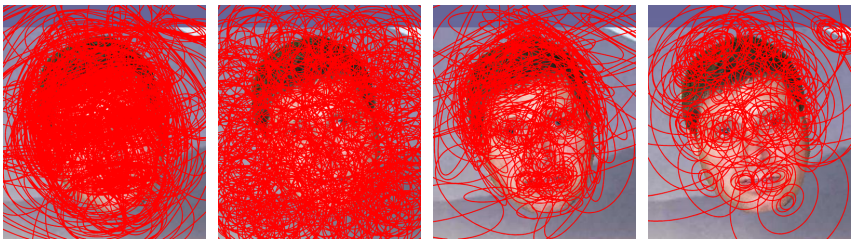
For the comparison we, selected the following detectors:

- *Harris-Laplace* Mikolajczyk and Schmid (2001).
- *Hessian-Laplace* Mikolajczyk et al. (2005b).
- *Harris-Affine* Mikolajczyk and Schmid (2002).
- *Hessian-Affine* Mikolajczyk et al. (2005b).
- *Maximally Stable Extremal Regions* (MSER) Matas et al. (2002).
- *Dense sampling* (see e.g. T.Tuytelaars, 2010).
- *Difference-of-Gaussian* (DoG) Lowe (2004).
- *Speeded-Up Robust Features* (SURF) Bay et al. (2006).

The detection results are demonstrated in Fig. 2.



(a) Harris-Affine (FS) (b) MSER (FS) (c) Dense sampling (d) Harris-Laplace (FS)

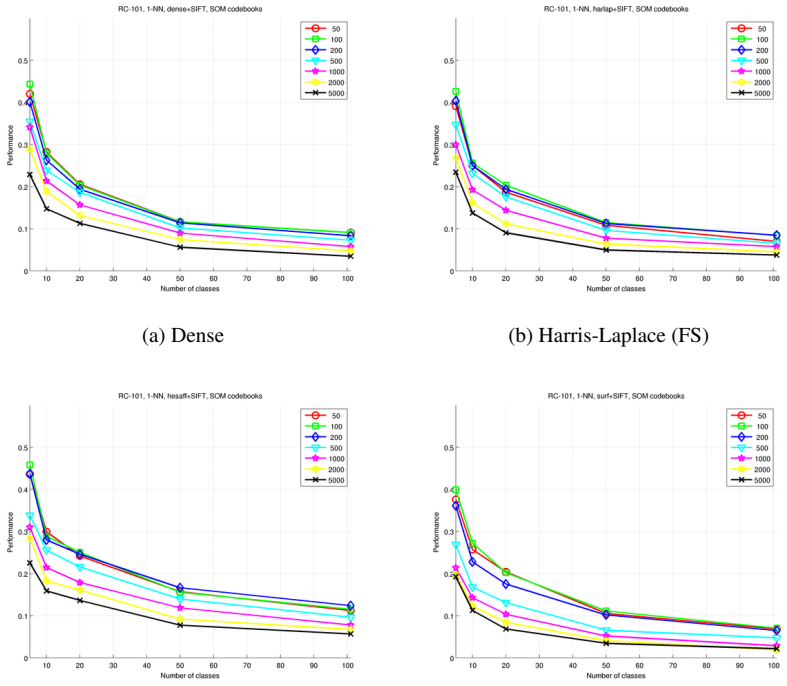


(e) Hessian-Affine (FS) (f) DoG (SIFT) (LV) (g) Harris-Laplace (LV) (h) Hessian-Laplace (LV)

Figure 2: Detected interest regions by using different methods and implementations. FS: implementation from the Feature Space web-site (Mikolajczyk et al. (referenced 2010)); LV: from the Lip-Vireo web-site (Zhao (referenced 2010)).

It is noteworthy that different implementations of the same detectors produce different results. For the comparison, we generated the codebooks with the SOM algorithm (see Sec. 3.2 for details) and utilised the nearest neighbour classification rule. The data

set used was randomized Caltech-101 (r-Caltech-101), which will be introduced in detail in Section 4. The performances of the four best methods are shown in Fig. 3. The Hessian-Affine detector (the Feature Space implementation) seems to provide the best results. For each detector method, the optimal codebook size was selected based on the previous experiment and the result graphs for each detector are shown in Fig. 4. This experiment verifies that Mikolajczyk's Hessian-Affine detector outperforms the other detectors in the 1-NN based VOC task. It is noteworthy, that the detector is not the same as in the original experiment (Mikolajczyk et al., 2005b), but the revised version Mikolajczyk et al. (referenced 2010).



(a) Dense

(b) Harris-Laplace (FS)

(c) Hessian-Affine (FS)

(d) SURF (FS)

Figure 3: (a)-(d) 1-NN classification accuracy for the SOM-generated codebooks of different sizes and for the four best-performing local feature detectors.

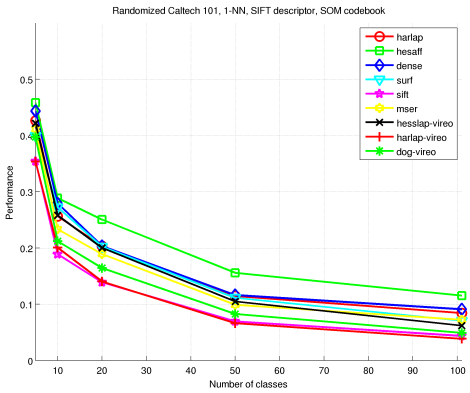


Figure 4: 1-NN classification accuracy for the SOM-generated codebooks of the optimal codebook sizes for each detector.

3.2. Codebook generation and feature construction

The most common approach to generate the codebook is to use a clustering technique. The baseline clustering method is the k-means algorithm (Csurka et al., 2004). K-means is known to have some weaknesses; for example, the cluster centres are typically found around high densities in data, and therefore, the input space is not evenly covered. Moreover, the visual codebook is not ordered, i.e. no special relationship between the neighbouring bins exists. To overcome these problems, Jurie and Triggs (2005) added a radius term to k-means in order to enforce the cluster centres to cover the input space better, and they also proposed a method to sort the codebook bins to make them meaningful. There are also many other codebook generation methods.

One family of algorithms for codebook generation are the ones typically used for data visualisation and exploration, such as the multi-dimensional scaling (MDS) (Borg and Groenen, 2005), Kohonen's self-organising map (SOM) (Kohonen, 1990), Isomap (Tenenbaum et al., 2000), and locally linear embedding (LLE) (Roweis and Saul, 2000). These methods have similar properties, and therefore, we select the one that can find a topological grouping of data points effectively: the self-organising map and its public implementation, the SOM Toolbox (Alhoniemi et al., 2000). The self-organising map has been successful compared to the k-means algorithm in our previous experi-

ments (Kinnunen et al., 2009). In generating the codebook, we used a one-dimensional SOM with a Gaussian neighbourhood and toroidal shape. The length of the SOM varied from 50 to 10 000. Fig. 5 illustrates how the detected local features are matched to the codebook for code assignment. In the figure, only the 20 best matching local features are shown to keep the image simple. The image feature for classification, the codebook histogram, is constructred by counting the matches per each code.

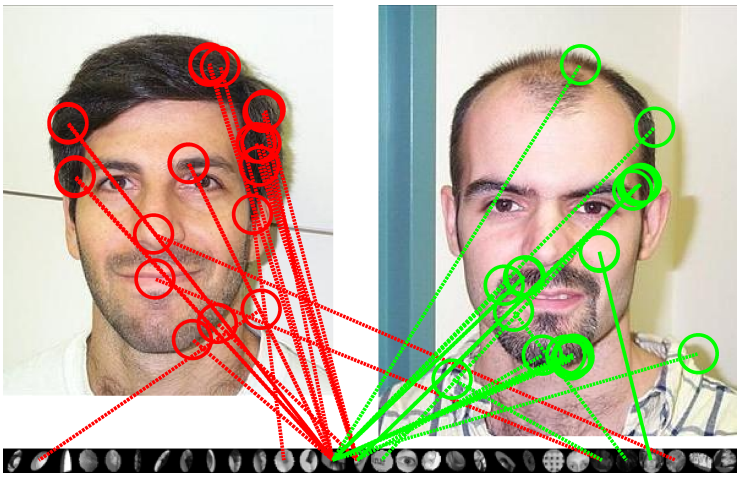


Figure 5: Example regions detected from two images of the same category and their code assignments in 1-D SOM codebook (the most representative regions in the training set demonstrate the codes).

3.3. Category discovery – "category book" – using the SOM algorithm

The unsupervised formation of the codebook is similar to the unsupervised discovery of object categories, and therefore, the same methods can be used. The main difference of our method as compared to Tuytelaars et al. (2010) is that the category book in our case is formed using the self-organising map.

One of the major contributions by Tuytelaars et al. (2010) was the comparison of codebook histogram normalisation methods. Tuytelaars et al. conducted experiments on two histogram normalisation methods, L1-norm and L2-norm normalisation, and a

few codeword normalisation methods: binarised-BoF, Term Frequency-Inverse Document Frequency (TF-IDF BoF) and Principal Compoment Analysis (PCA-BoF). In normalising the codeword, all bins of a certain code are normalised. In the binarised-BoF, the median of occurrences of each code is computed, and all bins below the median are set to zero, and all above to one. In TF-IDF, the number of occurrences of a code in an image (term frequency) is divided by the number of images containing the code (inverse document frequency). In the PCA-BoF, the histogram dimensionality is reduced to 20 by the principal component analysis (PCA). Tuytelaars et al. (2010) concluded that the L2-norm normalisation produces the best performance followed by the binarised-BoF. In Sec. 4, we repeat the experiments with the SOM codebook and category book to see whether these results are valid in our case.

In Fig. 6, we illustrate the SOM category book trained using 30 random examples from each of the 101 classes in Caltech-101. Each node in the 20×15 SOM map is represented by its best matching training image to illustrate self-organisation. For example, the cartoon characters dominate the left bottom corner, pandas and other furry animals are in the top left corner, faces are in the middle, and overall, neighbouring cells contain the same or similar classes. Note that sometimes the object background (e.g., "forest"), can dominate the class assignment.

4. Experiments

In this section, we report the results of three experiments. The first experiment demonstrates the importance of using randomised Caltech-101 (r-Caltech-101) (Kinnunen et al., 2010) instead of the original Caltech-101 set for the reliable evaluation of VOC methods. In addition, with the best detector, Hessian-Affine, our method achieves significantly better results than in the original study. The second experiment replicates the UVOC experiment by Tuytelaars et al. (2010) using the same 20 classes selected from Caltech-256. Our SOM-based method achieves similar performance, but as its advantage, it is less sensitive to the feature normalisation. The third experiment demonstrates the performance of our method in the unsupervised setting for r-Caltech-101.

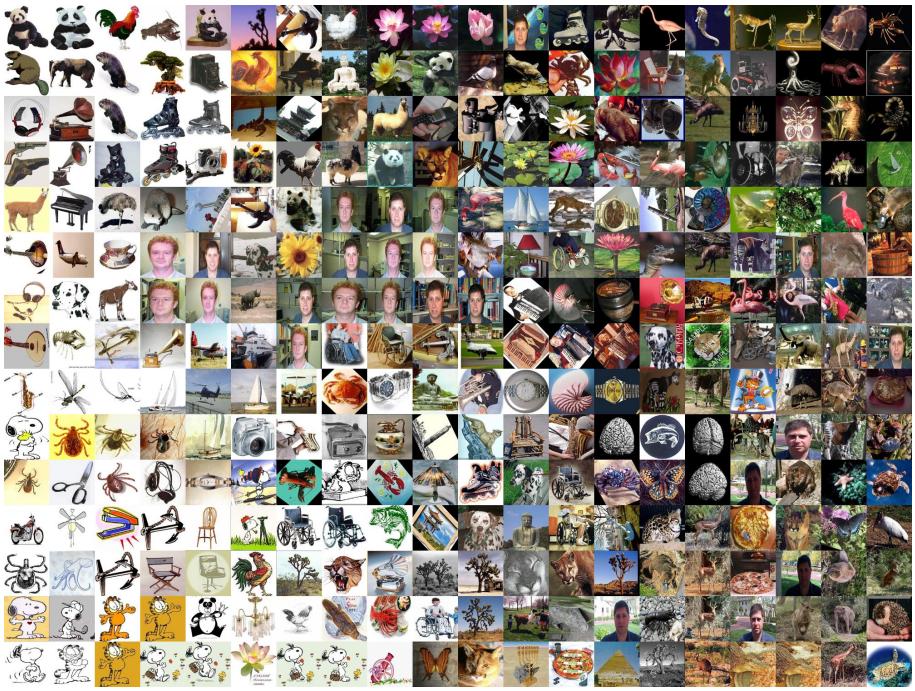


Figure 6: Caltech 101 (Fei-Fei et al., 2004) images categorized using the self-organising map. See Sec. 4.4 for the quantitative results.

4.1. Data

4.1.1. Caltech-101

Caltech-101 by Fei-Fei et al. (2004, 2006) contains roughly 8,600 images from 101 object categories. Each image contains only a single object, and the objects are cropped to the centre and rotated so that each object is roughly in the same pose. Some of the object categories are overlapping (e.g. *cougar body* and *cougar face*, *crocodile* and *crocodile head*, and *faces* and *faces easy*), but still the data set offers a very diverse collection of images. This is the most popular data set used in VOC evaluations.

4.1.2. Randomized Caltech-101 (r-Caltech-101)

The main disadvantage of Caltech-101 is that there is virtually no geometric variation of the objects in the images, and the backgrounds provide strong cues for certain

categories, such as the frequently occurring "book shelf" in the backgrounds of the faces class. These issues were reported by Ponce et al. (2006), and motivated by the difficulties in method development due to the database bias, we proposed an artificially generated randomised Caltech-101 (r-Caltech-101) (Kinnunen et al., 2010). In r-Caltech-101, the backgrounds are replaced with random landscape images, and the objects (foregrounds) are translated, scaled, and rotated randomly. The foreground segments were provided with the original data (Fei-Fei et al., 2006). The randomisation process makes learning more difficult due to pose variation and removal of the undesired background bias of the original data.

In the experiments, we randomly chose 5 to 101 classes, and from each class, 30 random images for training and 20 for testing. For a few of the classes, there are less than 20 images for the test set. Our evaluation method measures performance as a function of the number of classes to evaluate how well the method scales up, and to make optimisation for any particular data set more difficult. Data set partitioning to a training and testing set was suggested already by Fei-Fei et al. (2004). The performance of the system is measured as the average class-wise classification accuracy. This makes each category equally important despite the fact that some of the categories have more images than others.

4.1.3. Caltech-256

Caltech-256 (Griffin et al., 2007) is a newer data set from the authors of Caltech-101. It contains roughly 30,000 images from 256 object categories. Some of the categories are identical with Caltech-101, but generally, the objects are not centered or rotated to any standard pose, as with Caltech-101. As an extra challenge, the images are captured from various viewpoints, which causes difficult 3-D variation. In our experiments, we used the same 20 categories as Tuytelaars et al. (2010).

4.2. Experiment 1: Caltech-101 vs. r-Caltech-101

We repeated the experiment originally presented in the work by Kinnunen et al. (2010) where the performances were computed for different parts of the Caltech-101 and r-Caltech-101 images: Caltech-101 images (full), r-Caltech-101 images (full), Caltech-

101 backgrounds only (Bg), r-Caltech-101 backgrounds only (Bg), original Caltech-101 foregrounds (Fg) and new r-Caltech-101 foregrounds (Fg). We used the same image set, e.g. r-Caltech-101 foregrounds, in both training and testing phases. The performance curves for all these cases are plotted in Fig. 7 and the numbers given in Table 1, where the results obtained using the Hessian-Affine detector are notably better than those by Kinnunen et al. (2010), where Lowe's SIFT detector was used.

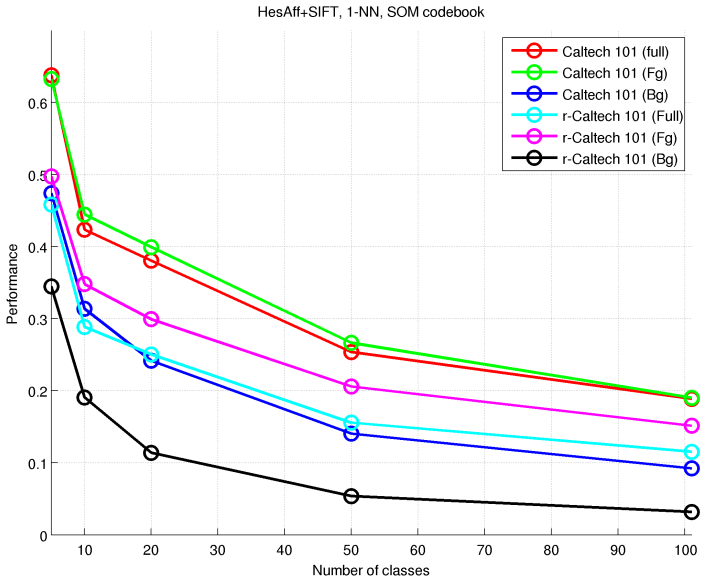


Figure 7: Experiment 1: 1-NN classification performances for data generated from Caltech-101 and r-Caltech-101.

The best performance (Fig. 7 & Table 1) was achieved using the foregrounds only from the original Caltech-101 (the green curve). The background clutter had virtually no effect on the performance with Caltech-101 since the performance with the full images was almost the same (the red curve). The rotation and scaling of the foregrounds affected the detected features, which is evident from the results for the r-Caltech-101 foregrounds (the magenta curve), which is the third best, but clearly outperformed by the original foregrounds and full images. In r-Caltech-101, the background clutter had

Table 1: Experiment 1: Caltech-101 vs r-Caltech 101 results.

	C101 Full		C101 Fg		C101 Bg		RC101 Full		RC101 Fg		RC101 Bg	
#categ.	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
5	**0.638**	0.079	0.632	0.063	0.474	0.130	0.458	0.090	0.497	0.117	0.345	0.065
10	0.423	0.057	**0.445**	0.067	0.314	0.050	0.289	0.042	0.348	0.046	0.191	0.045
20	0.380	0.058	**0.399**	0.060	0.242	0.039	0.250	0.039	0.299	0.047	0.114	0.020
50	0.254	0.038	**0.266**	0.022	0.140	0.019	0.156	0.020	0.206	0.031	0.054	0.008
101	0.189	0.000	**0.190**	0.000	0.092	0.003	0.115	0.000	0.151	0.000	0.032	0.001

the expected result as it significantly reduced the performance as compared to the r-Caltech-101 foregrounds. Interestingly, the Caltech-101 background only (the blue curve) achieved almost the same performance as the full r-Caltech-101 images (the cyan curve). The worst performance was achieved with the r-Caltech-101 backgrounds only (the black curve). It is noteworthy that the worst result does not correspond to random chance, which can be explained by the fact that since the features on the foreground were just omitted, the total amount of detected features correlates with the object sizes, and therefore, provides a cue of the class.

As a summary, the r-Caltech-101 data set provides a more challenging test bench for the VOC methods, since the background clutter and invariance have a drastic effect on the performance. The r-Caltech-101 does not provide natural data, but it should be used with Caltech-101 to represent how well a method can tolerate geometric transformations and background clutter.

4.3. Experiment 2: Comparison to the state-of-the-art

In this experiment, our approach was compared to that of Tuytelaars et al. (2010), which represents the current state-of-the-art in UVOC. Our results are reported for the same 20 categories of Caltech-256. For unsupervised categorisation, we replaced the previously used 1-NN classifier with the category book generated by the SOM algorithm. For comparison, we also conducted the same experiments with the neural gas (Martinetz and Schulten, 1991), k-means algorithms, Normalised Cuts (Shi and Malik, 2000) and k-Means with Kernel-PCA (Tuytelaars et al., 2010). The performance is reported by computing the conditional entropy defined in Eq. 5. In this

experiment, we additionally investigated the effect of feature normalisation.

In Fig. 8, conditional entropy graphs are shown for the different methods and sizes of the codebook. In the Tuytelaars et al. protocol, the size of the category book was fixed to the number of categories. The different colours denote the different methods and the markers denote the different normalisation methods. The numerical values are shown in Table 2.

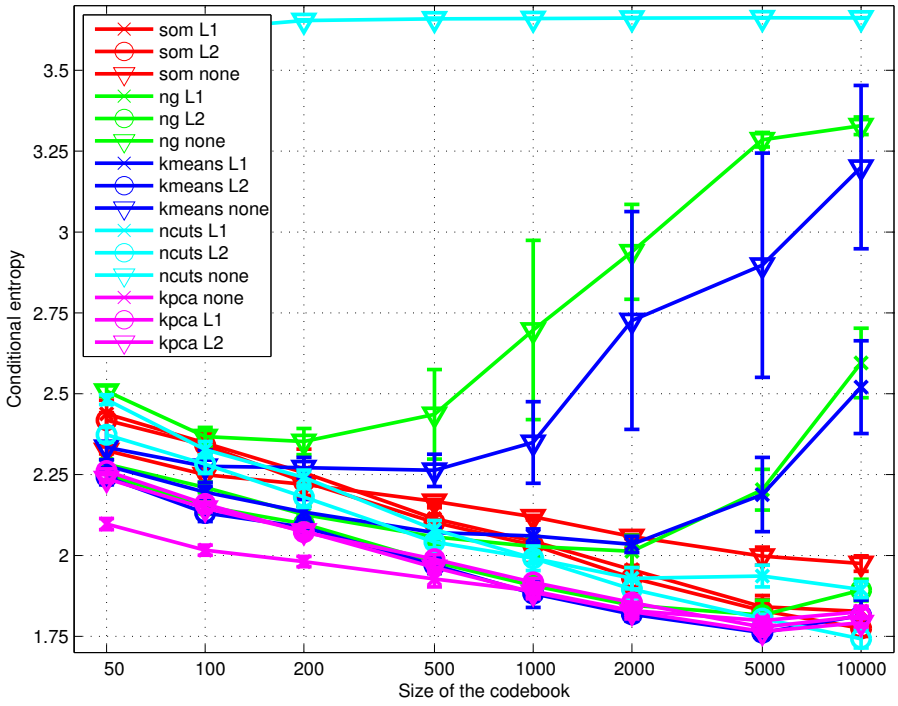


Figure 8: Experiment 2: Performances in the UVOC experiment with the 20 Caltech-256 classes (Tuytelaars et al., 2010). The red line stands for SOM categorisation, the green for Neural Gas and the blue for k-Means and cyan for Normalised Cuts. The cross denotes L1-normalisation, the circle L2-norm normalisation, the triangle denotes that codebook histograms are not normalised.

Two important findings can be made from the results (Fig. 8 & Table 2). First, the large codebooks provide better results. Second, the k-means, neural gas and Nor-

Table 2: Experiment 2: Comparison to the state-of-the-art results.

Codebook size	Categorisation method	Normalisation					
		None		L1		L2	
		CE	std	CE	std	CE	std
100	Self-Organising Map	2.25	0.01	2.35	0.03	2.34	0.04
500		2.17	0.01	2.11	0.04	2.10	0.05
2000		2.06	0.01	1.96	0.07	1.93	0.04
10000		1.98	0.02	1.83	0.04	1.77	0.02
100	Neural Gas	2.37	0.03	2.21	0.02	2.15	0.02
500		2.44	0.14	2.06	0.02	1.98	0.02
2000		2.94	0.15	2.01	0.05	1.85	0.02
10000		3.33	0.03	2.60	0.11	1.89	0.03
100	K-Means	2.28	0.05	2.20	0.02	2.13	0.03
500		2.26	0.05	2.07	0.02	1.97	0.03
2000		2.73	0.34	2.03	0.03	1.82	0.02
10000		3.20	0.25	2.52	0.14	1.81	0.05
100	Normalised Cuts	3.63	0.00	2.33	0.03	2.28	0.03
500		3.66	0.00	2.08	0.03	2.04	0.02
2000		3.66	0.00	1.93	0.03	1.90	0.03
10000		3.66	0.00	1.89	0.02	1.74	0.03
100	Kernel PCA	2.02	0.02	2.16	0.02	2.15	0.01
500		1.93	0.02	1.99	0.01	1.96	0.01
2000		1.83	0.03	1.85	0.02	1.83	0.01
10000		1.83	0.02	1.81	0.02	1.79	0.01

malised Cuts algorithms are very sensitive to the data normalisation, whereas SOM and Kernel PCA are not. Moreover, the performance of SOM and Normalised Cuts steadily increases. In the original work by Tuytelaars et al. (2010), the best performance was achieved with dense sampling, binarised features, and a k-means category book. A comparable performance was achieved with our method by using the Hessian-Affine detector, SIFT, 10000 word SOM codebook, L2-normalised features and SOM image categorisation.

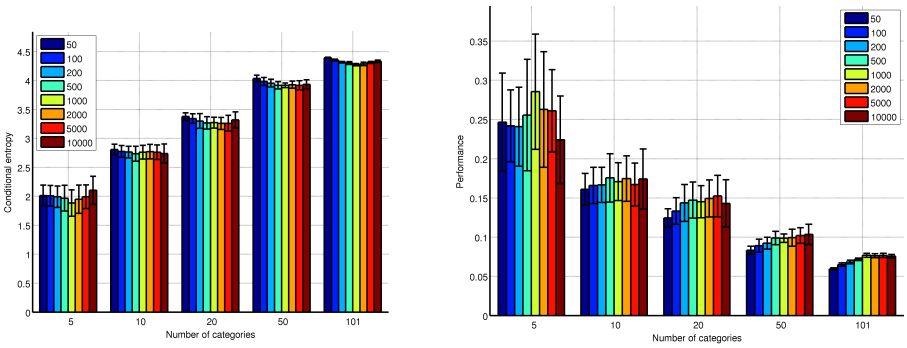
As a summary, we can conclude that this experiment verified our previous indicating results that the SOM algorithm is a competitive alternative to clustering methods, such as the k-means, k-means with Kernel PCA and Normalised Cuts algorithm. It also has some advantageous properties, such as its tolerance to data normalisation.

4.4. Experiment 3: Unsupervised object discovery from r-Caltech-101

This experiment is the most challenging with the images from the r-Caltech-101 data set. For each iteration, we randomly select 30 images from each category, and following the previous experiment, fix the codebook size to the true number of categories. The other parameters are selected based on the previous experiments: the Hessian-Affine detector, SIFT descriptors, and the L2-norm normalisation of the histogram features.

We report both performance measures, the conditional entropy by Tuytelaars et al. and our classification accuracy (Fig. 9 & Table 3). Note that for the conditional entropy, the smaller values are better, and for the classification accuracy, the greater values are better. By comparing the results in Figs. 9a and 9b, it is obvious that the both performance measures provide the same information: the performance steadily degrades as the number of categories increases, and on average, the codebook size 1000 provides the best performance, however, without significant difference to others.

In our opinion, the classification accuracy performance in Fig. 9b is more intuitive. For example, it reveals that the performance is only slightly better than pure chance (0.2 for five classes, 0.1 for 10 classes, etc.). Interestingly, the performance degrades as the number of categories increases, as is expected, but compared to pure chance,



(a) (b)

Figure 9: Experiment 3: Results for the r-Caltech-101 UVOC experiment: (a) conditional entropy; (b) classification accuracy.

the performance improves (approx. $2\times$ better for 20 categories and $5\times$ better for 100 categories).

The problem of this experiment is in the performance measures of both Tuytelaars et al. and us, which require the size of the category book to be fixed to the number of categories. This avoids the problem of model selection, but for methods such as SOM, the optimal size cannot be expected to correspond to the number of categories. To investigate this limitation, we adopted the "oracle" approach (Tuytelaars et al., 2010), which requires separate training and test sets. Each SOM cell represented the category which dominates it. Results for the various sizes of the category book and for all of the 101 r-Caltech-101 categories are shown in Fig. 10. The best performance (0.10) was achieved with the category book size of 1000. Note that the codebook size of $30 \times 101 = 3030$ would, in principle, correspond to the 1-NN classification rule.

As a summary, we conclude that the UVOC performance significantly decreases compared to the baseline VOC results in Fig. 3 (the cyan graph). The UVOC performance can be improved by better model selection. Moreover, we claim that the conditional entropy performance proposed by Tuytelaars et al. (2010) could be replaced with the direct classification accuracy performance which provides the same interpre-

Table 3: Experiment 3: Unsupervised object discovery from r-Caltech-101 results.

Codebook size	100		500		2000		10000	
#categories	mean	std	mean	std	mean	std	mean	std
	Classification accuracy							
5	0.242	0.046	0.256	0.071	0.263	0.074	0.224	0.056
10	0.166	0.023	0.175	0.031	0.175	0.029	0.174	0.038
20	0.133	0.017	0.147	0.023	0.149	0.024	0.143	0.030
50	0.089	0.008	0.099	0.009	0.099	0.011	0.103	0.013
101	0.065	0.002	0.072	0.002	0.076	0.002	0.076	0.002
	Conditional entropy							
5	2.008	0.178	1.966	0.223	1.949	0.244	2.102	0.243
10	2.775	0.100	2.733	0.128	2.770	0.126	2.737	0.166
20	3.340	0.082	3.266	0.107	3.258	0.105	3.318	0.140
50	3.982	0.069	3.916	0.062	3.926	0.061	3.932	0.077
101	4.351	0.014	4.296	0.024	4.287	0.026	4.325	0.020

tation, but is much more intuitive. Moreover, fixing the size of the category book, i.e. the number of "clusters", does not provide reliable method comparison, but the oracle approach should be used, and the methods should perform independently and freely make the class assignments.

5. Conclusion and future work

In this paper, we investigated the recent problem of unsupervised visual class discovery. We presented a new approach based on the self-organisation principle making use of the self-organising map algorithm. Our method achieved performance comparable to the state-of-the-art and has certain advantageous properties, such as robustness against data normalisation. The building blocks of our approach, i.e. the SOM codebook and category book, local feature detector, and feature normalisation, were selected through carefully designed and executed experiments presented in this paper. We also proposed and demonstrated a more suitable and intuitive performance measure as compared to the previously proposed conditional entropy.

Future work will address the problem of unsupervised model selection, which

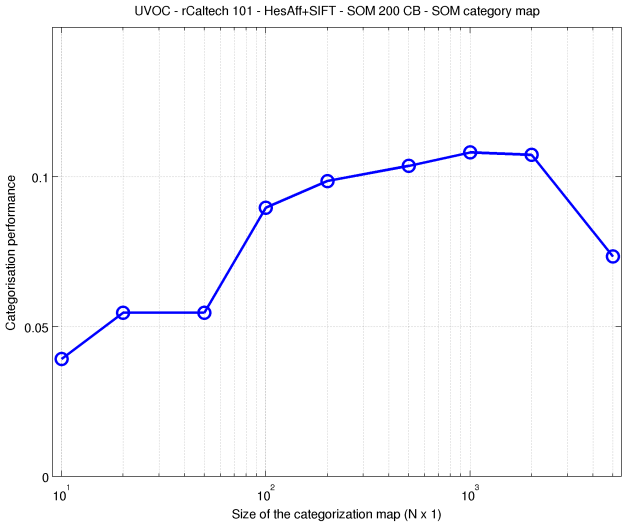


Figure 10: Results for all the 101 categories using various category book sizes.

should be used to select the category books best representing the data. The same problem concerns also the selection of the optimal codebook size. In addition, novel approaches including the use of spatial information will be investigated.

Acknowledgements

The authors wish to thank the Academy of Finland and the partners of the VisiQ project (no. 123210) for their support (URL: http://www2.it.lut.fi/project/visiq/).

Alhoniemi, E., Himberg, J., Parhankangas, J., Vesanto, J., 2000. SOM Toolbox. http://www.cis.hut.fi/somtoolbox/. 11

Bar-Hillel, A., Weinshall, D., 2008. Efficient learning of relational object class models. Int J Comput Vis 77, 175–198. 2

Bart, E., Porteous, I., Perona, P., Welling, M., 2008. Unsupervised learning of visual taxonomies. In: CVPR. 3

Bay, H., Tuytelaars, T., Gool, L., 2006. Surf: Speeded up robust features. In: In ECCV. pp. 404–417. 9

Biederman, I., 1987. Recognition-by-components: A theory of human image understanding. Psychological Review 94(2), 115–147. 4

Borg, I., Groenen, P., 2005. 2nd Edition. New York: Springer. 11

Burl, M., Leung, T., Perona, P., 1996. Recognition of planar object classes. In: CVPR. 3

Csurka, G., Dance, C., Willamowski, J., Fan, L., Bray, C., 2004. Visual categorization with bags of keypoints. In: ECCV Workshop on Statistical Learning in Computer Vision. 2, 11

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A., 2008. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html. 2, 3

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A., 2009. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. 2, 3

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A., 2010. The PASCAL Visual Object Classes (VOC) challenge. Int J Comput Vis 88 (2), 303–338. 3, 7

Fei-Fei, L., Fergus, R., Perona, P., 2004. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: CVPR Workshop on Generative-Model Based Vision. 14, 15

Fei-Fei, L., Fergus, R., Perona, P., 2006. One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. 28 (4), 594. 2, 3, 14, 15

Griffin, G., Holub, A., Perona, P., 2007. Caltech-256 object category dataset. Tech. Rep. 7694, California Institute of Technology.
URL http://authors.library.caltech.edu/7694 2, 3, 15

Holub, A., Welling, M., Perona, P., 2005. Exploiting unlabelled data for hybrid object classification. In: NIPS Workshop in Interclass Transfer. 2

Jurie, F., Triggs, B., 2005. Creating efficient codebooks for visual recognition. In: ICCV. pp. 604–610. 11

Kinnunen, T., Kamarainen, J.-K., Lensu, L., Kälviäinen, H., 2009. Bag-of-features codebook generation by self-organization. In: International Workshop on Self-Organizing Maps (WSOM). 12

Kinnunen, T., Kamarainen, J.-K., Lensu, L., Lankinen, J., Kälviäinen, H., 2010. Making visual object categorization more challenging: Randomized caltech 101 data set. In: International Conference in Pattern Recognition (ICPR). 13, 15, 16

Kohonen, T., September 1990. The self-organizing map. Proc. of the IEEE 78 (9), 1464–1480. 3, 4, 11

Kohonen, T., Kaski, S., Lagus, K., Honkela, J., 1996. Very large two-level SOM for the browsing of newsgroups. In: Int. Conf. on Artificial Neural Networks (ICANN). pp. 269–274. 4

Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., Saarela, A., 2000. Self organization of a massive document collection. IEEE Trans. on Neural Networks 11 (3). 4

Laaksonen, J., M. Koskela, S. L., Oja, E., 2000. Picsom - content-based image retrieval with self-organizing maps. Pattern Recognition Letters 21 (13-14). 4

Li, F., Carreira, J., Sminchisescu, C., 2010. Object recognition as ranking holistic figure-ground hypotheses. In: CVPR. 2

Lowe, D., January 2004. Distinctive image features from scale-invariant keypoints. Int J Comput Vis 20, 91–110. 9

Martinetz, T., Schulten, K., 1991. A "Neural-Gas" Network Learns Topologies. Artificial Neural Networks I, 397–402. 17

Matas, J., Chum, O., Urban, M., Pajdla, T., 2002. Robust wide-baseline stereo from maximally stable extremal regions. In: Proc. of the British Machine Vision Conf. pp. 384–393. 9

Mikolajczyk, K., Leibe, B., Schiele, B., 2005a. Local features for object class recognition. In: CVPR. 8

Mikolajczyk, K., Schmid, C., 2001. Indexing based on scale invariant interest points. In: ICCV. pp. 525–531. 9

Mikolajczyk, K., Schmid, C., 2002. An affine invariant interest point detector. In: ECCV. pp. 128–142. 9

Mikolajczyk, K., Schmid, C., 2005. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis & Machine Intelligence 27 (10), 1615–1630. 8

Mikolajczyk, K., Tuytelaars, T., Matas, J., Schmid, C., Zisserman, A., referenced 2010. Featurespace. http://www.featurespace.org/. 9, 10

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L. V., 2005b. A comparison of affine region detectors. Int J Comput Vis 65 (1/2), 43–72. 8, 9, 10

Ponce, J., Berg, T., Everingham, M., Forsyth, D., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B., Torralba, A., Williams, C., Zhang, J., Zisserman, A., 2006. Dataset issues in object recognition. In: Workshop on Category Level Object Recognition. pp. 29–48. 15

Roweis, S., Saul, L., 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290 (5500), 2323–2326. 11

Russell, B., Torralba, A., Murphy, K., Freeman, W., 2008. Labelme: A database and web-based tool for image annotation. Int. J. of Comp. Vision 77 (1-3), 157–173. 2

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22 (8), 888 –905. 17

Sivic, J., Russell, B. C., Zisserman, A., Freeman, W. T., Efros, A. A., 2008. Unsupervised discovery of visual object class hierarchies. In: Proc. of the Computer Vision and Pattern Recognition. pp. 1–8. 3, 5, 7

Tenenbaum, J., de Silva, V., Langford, J., 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290 (5500), 2319–2323. 11

T.Tuytelaars, 2010. Dense interest points. In: CVPR. 9

Tuytelaars, T., Lampert, C., Blaschko, M., Buntine, W., 2010. Unsupervised object discovery: A comparison. Int J Comput Vis 88 (2). 2, 3, 4, 5, 7, 12, 13, 15, 17, 18, 20, 21

Weber, M., Welling, M., Perona, P., 2000. Unsupervised learning of models for recognition. In: ECCV. 2

Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C., 2006. Local features and kernels for classification of texture and object categories: A comprehensive study. Int. J. of Computer Vision 73 (2). 8

Zhao, W., referenced 2010. LIP-VIREO local interest point extraction toolkit. http://vireo.cs.cityu.edu.hk. 9