Compact Description of Local Features

Iris HEISTERKLAUS[1]*, Christopher BULLA*[1]

[1]Institute of Communications Engineering, RWTH Aachen University, 52056 Aachen, Germany

{heisterklaus, bulla}@ient.rwth-aachen.de

Abstract. *Considering the growing amount of digital images in all kind of databases, the search for specific content remains a problem. Content-based image retrieval based on local features is a promising approach but comes with the problem of being memory and computational intensive. The* bag of keypoints *approach reduces the feature vectors to one histogramm per image. This paper shows an efficient clustering strategy to obtain the codebook needed for the* bag of keypoints *computation. With the codebook, compact image descriptors are computed. An object classification with compact descriptors is conducted to test the performance of the obtained compact descriptors. This study shows that multi-class classification with up to 15 classes is possible.*

Keywords

Local features, SIFT, clustering, object classification, SVM, bag of keypoints, content-based image retrieval

1. Introduction

Due to the advances in digital photography the number of images in publicly available databases, especially on the internet, has been growing tremendously. Current search strategies are mainly text-based and use tags assigned to the images. The problem remains that image content is far too complex to be described by some tags and there is an abundance of images that are not annotated and cannot be found using this search strategy. Local features are a way of describing image content that addresses this problem. For distinctive keypoints of an image, descriptors are computed which describe the area around these keypoints. The number of computed features depends on the image content. For complex images, up to thousands of features may be found. To be able to use local features in a search engine for an image database, the feature vectors for every database image need to be computed and stored. Depending on the number of images, this can cause hugh memory requirements. For retrieval, a query image is given to the search engine and the local features for this image are computed. Those features are compared to all the stored features in the database. This procedure is computationally intensive due to the many comparative operations needed.

A compact feature based description can solve this problem. This paper shows a method to describe images with a single *bag of keypoints* vector thus reducing the needed memory for storing the image features. At the same time, less comparations are needed for comparing images as every image is only described by a single vector. To obtain the *bag of keypoints* a clustered feature space is needed. A tree-based strategy for clustering the features space that also provides easy access to the computed cluster centers for use in the *bag of keypoints* computation is introduced. To test if the *bag of keypoints* provides distinctive information about the image content, an object classification with multiple classes is done and evaluated.

Section 2 addresses local features and describres SIFT as an example. Section 3 describes the compact description of images. The features space clustering which is needed for the compact image description shown in section 4. An object classification experiment is described in section 5 and the results are presented in section 6. Section 7 summarizes the findings.

2. Local features with SIFT

Local features are salient image keypoints or regions which provide rich local information about the image. To find the position of the keypoints, local interest point detectors are applied and the found interest points are described by a feature descriptor. The choice of the interest point detector and descriptor are application dependant. A survey of interest point detectors can be found in [1].

In this paper, the *Scale Invariant Feature Transform* (SIFT) by Lowe [2] is used to compute keypoints and corresponding descriptors. SIFT has proven to be efficient and often outperforms other descriptors. The computed features are scale and rotational invariant. They are also robust against local affine distortions, as well as illumination and viewpoint changes. This makes them a good choice for describing objects in different images as the position, scale and the environmental conditions may change between different images. SIFT has four major steps:

1. Scale space extrema detection

   Potential interest points are detected using a Difference of Gaussian function over all scales and image locations. The scale space representation $L(x, y)$ of an image $I(x, y)$ is computed by convolution with a Gaussian kernel where σ is the standard deviation:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \\ \text{with } G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

The *Difference of Gaussian* function convolved with the image can be computed from the difference of two nearby scales:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma)) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (2)$$

The two scales are separated by a constant multiplicative factor k. Figure 1 shows the computation of the scale space images for two octaves and the DoG computation from the different Gaussians. Within the DoG images, the extrema are computed which constitute the detected interest points. To further refine the positions of the keypoints, the detected extremas are compared to their 26 neighbours in the current and adjacent scales. The exact position of the minimum or maximum is found by interpolation.

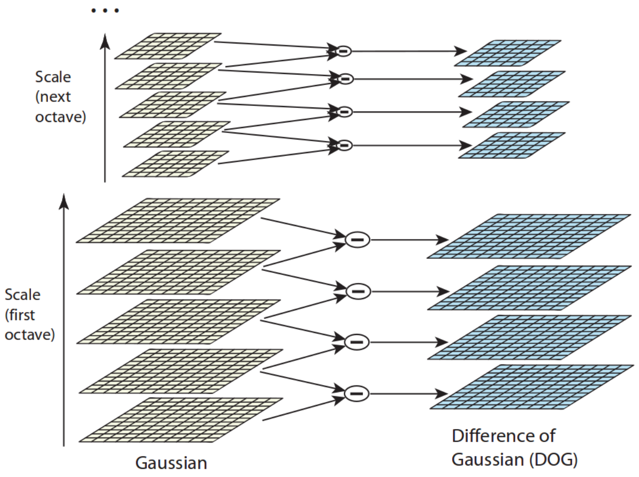


Fig. 1: Computation of scale space images, from [2]. On the left, the computation of Gaussian blurred images at various scales is shown. From these, the Difference of Gaussian images on the right are computed which are then used for interest point detection.

2. Keypoint localization

   Image regions with low contrast provide unstable interest points. The contrast of the image is evaluated and points that provide low contrast are discarded. Also, interest points that are located along an edge tend to get unstable. The elimination of these points reduces the sensitivity to noise.

3. Orientation assignment

   A consistent orientation based on the image gradients is assigned to each candidate point which provides invariance to rotation. The Gaussian smoothed image $L(x, y)$ at the closest scale to the keypoint is selected and the gradient magnitude

$$\begin{aligned} m(x, y) =& [(L(x+1, y) - L(x-1, y))^2 \\ &+ (L(x, y+1) - L(x, y-1))^2]^{\frac{1}{2}} \end{aligned} \quad (3)$$

   and orientation

$$\phi(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right) \quad (4)$$

   are computed using pixel differences.

4. Keypoint descriptor

   The keypoint descriptor is built from the image gradients as an orientation histogram. It is accumulated from the gradient orientations and each sample that is added is weighted by its gradient magnitude and by a gaussian-weighted circular window. Orientation histograms over subregions of 4×4 patches are built. Each histogram has 8 bins which leads to a $4 \times 4 \times 8 = 128$-dimensional descriptor vector. Figure 2 shows the computed gradients and orientations for the image patches with an example of a division into 2×2 patches. The circle indicates the regions of the Gaussian weights. The orientation histograms of the subregions are shown on the right side. The length of the arrows corresponds to the sum of the gradient magnitudes.

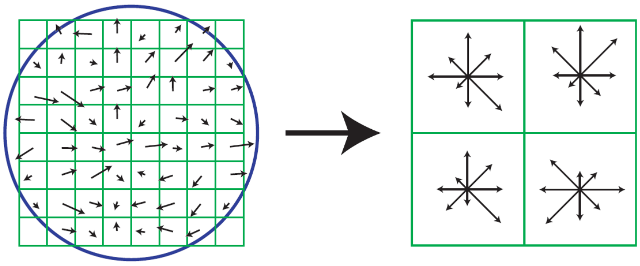


Fig. 2: From the Gaussian weighted image gradients (left), the corresponding orientation histogram for a division in 4 subpatches is computed (right), from [2].

3. Compact description of images

For compact image description, the *bag of keypoints* [3] approach was used. The bag of keypoints model originates in document classification where it is known as *bag of words*. A predefined dictionary, also called a codebook, is used to determine to which class a document belong. The document is analyzed regarding the occurence of words within the codebook and dependent on this the classification is done.

In computer vision, this model treats the feature vectors as *visual words* and assigns them to an entry in a predefined codebook. The codebook contains specific vectors and each occuring feature vector is mapped on the nearest codebook entry. Then a histogram of the feature vectors within the image is build which describes the whole image. Images that contain the same object are expected to have similar feature vectors thus generating similar histograms. This histogram is the *bag of keypoints*. Figure 3 shows how a bag of keypoints is computed for an input image. Firstly, the descriptors Y for the input image are computed. Then, they are mapped to the corresponding codebook entries to get the descriptors Y'. The histogram of all descriptors Y' is computed and builds the *bag of keypoints*.

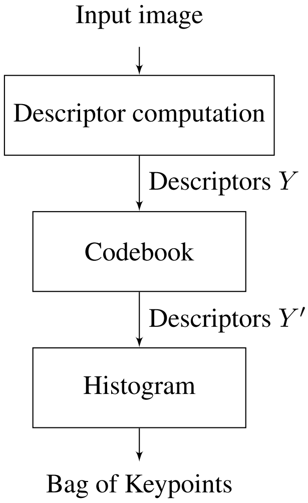


Fig. 3: *Bag of keypoints* computation for a single image. Descriptors vectors are computed and mapped to a codebook. All mapped vectors build a histogram, the *bag of keypoints* vector.

4. Feature space clustering

The SIFT feature space is 128-dimensional so there are 256^{128} possible different feature vectors. The feature space clustering builds groups of vectors. Every feature vector is assigned to a cluster which is represented by its center. By this, a codebook that maps every occuring feature space vector to a specific codebook entry is obtained and can be used for the bag of keypoints computation.

To get an impression of the distribution of feature vectors in the feature space, the MIRFLICKR database [4] was used. This database contains 25000 images from flickr.com and images were selected to provide a representative image collection. For all images in the MIRFLICKR database the SIFT features were computed. During this process, 15.958.172 feature points were computed. This is an average of 638 features points per image but the actual number of features per image varies greatly. As the database contains a representative selection of images, the computed feature vectors give an adequate impression of the descriptor distribution for image feature vectors in general.

In the next step, the computed features are clustered to get a codebook of a predefined size. The feature space analysis shows that features are not uniformly distributed but are clustered around the origin of the 128-dimensional space. We assume that feature vectors from all images will behave in a similar manner. Accordingly, feature vectors near the origin will appear more often than others. The clustering algorithm takes care that all cluster are approximately the same size. Thus, regions with higher feature vector density get more clusters than others.

A hierarchical k-means clustering approach is used [5]. This can be efficiently implemented using a binary tree. Starting with one giant cluster, this cluster is split in two new clusters using k-means. From now on the largest cluster is split again using k-means until the predefined number of clusters is reached. The binary tree leaves contain the centers of the clusters which constitute the codebook.

In a regular codebook, a computed feature vector has to be compared to all codebook entries to find the corresponding cluster. The use of the tree reduces the amount of comparisions needed. At every tree node, the branch with the nearest cluster center is chosen and the leaf containing the corresponding cluster center is found within a small number of comparisons depending on the current tree depth.

5. Object classification using compact features

It is an easy task for a human to decide whether an object is present in an image even if the object is partially occluded, the lighting conditions are different or the object has a different orientation. For a computer this task is highly challenging. Local features address this problem quite well. This section analyzes if the compact descriptors are still able to classify images according to their content.

A classifier decides on basis of specific features of an observation to which of a group of known classes the observation belongs. A support vector machine (SVM) [8] was used for the object classification. A SVM is a non-probabilistic binary linear classifier which uses supervised learning. Being a binary classifier, the standard SVM is only able to solve two class classifications. The two most popular strategies for multi-class SVM are *One against One* and *One against the Rest*. *One against One* trains $k(k-1)/2$ classifiers, one SVM for each pair of classes with k being the number of classes. The classification is done in a *max votes win* strategy, so each sample is assigned to the class it got most votes from. *One against the Rest* trains one SVM per class so each classifier distinguishes the samples of one class from the samples of all other classes. With both strategies inconsistent classification results can occur if one class gets the same number of labels for two or more classes. The SVM is also able to perform nonlinear classification using the kernel trick where inputs are mapped into a feature space of higher dimension [6].

The first part of classification using a SVM is a learning phase. The SVM is given labeled instances of the different classes to learn a classificator model. After the learning phase comes the test phase were classes are predicted for samples and compared to the actual labels of the samples. The division into two different learning and test datasets is crucial as an overfitting to the training data can occur. Using a different set of data for testing, this overfitting can be discovered.

The used test database is the Caltech-256 object database [7]. 15 classes containing more than 200 images were used to have sufficient images for training and test. For every image the bag of keypoints descriptor was computed and stored. The data was divided into training and test datasets. Each training set contains 150 images and each test set contains 50 images. 20 arbitrary splits of the whole dataset into test and training data were generated to perform a statistical evaluation.

6. Results

Object classification was performed with two classes, five classes and 15 classes. For each of the 20 generated data sets a classification was performed and the mean and standard deviation of the classifier performance was computed for each class. The evaluation of the classification results was done by precision-recall plots. Precision describes the subset of correctly labeled samples of the set that got the same label from the classifier.

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}. \qquad (5)$$

Recall is the subset of correctly labeled samples of the whole set of that class.

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}. \qquad (6)$$

The nearer the results are to the value 1, the better the result. Precision and recall were computed on the mean and standard deviation over all conducted experiments.

Figure 4 shows the result for a two class classification. For both classes, values near to 1 are reached. A classification with two object classes using compact descriptors is possible and yields very good results.

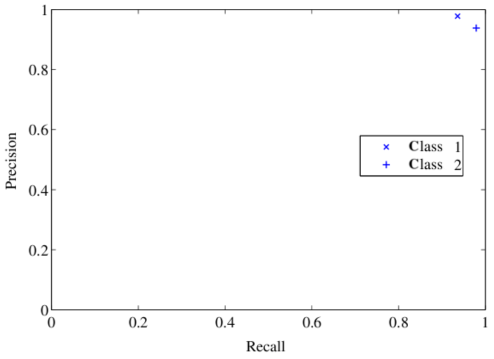


Fig. 4: Classification result for 2 classes.

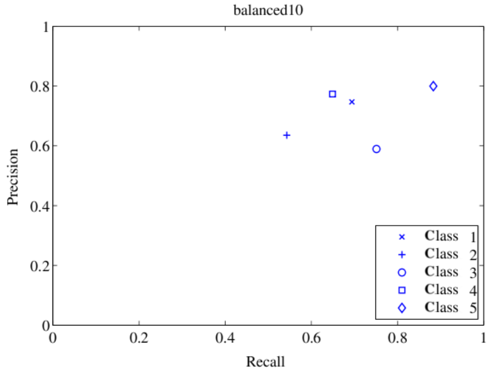


Fig. 5: Classification result for 5 classes.

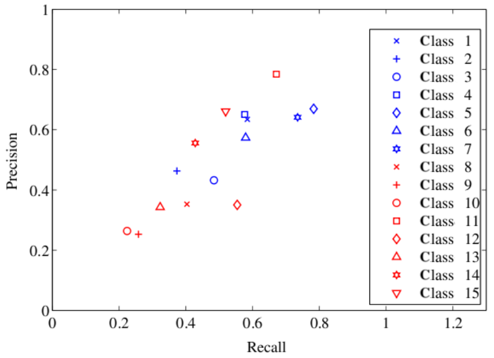


Fig. 6: Classification result for 15 classes.

Figure 5 shows the results for a classification using five object classes. The general result is not as good as the result obtained in two class classification. All classes reach precision and recall values higher than 0.5 which is still a good result for a five class classification. Class 1 and Class 2 from the two class classification are included as Class 1 and Class 2 in this experiment. A deterioration of the obtained precision and recall values for this two classes compared to the two class classification is observed. Table 1 shows the confusion matrix for this classification result. The confusion matrix shows the total number of images assigned to a class, here again computed from mean values. On the diagonal, the correct predicted imags are shown, all other matrix entries show false predictions. It can be observed that Class 3 with 38.1 and Class 5 with 45.7 correctly assigned images obtain the best result. Noticeable is the difference in the falsely assigned images, Class 5 has 11.7 wrong predictions whereas Class 3 has 29.0 wrong predictions. The precision-recall plot also shows this behaviour, Class 5 has precision and recall values around 0.8 whereas Class 3 reaches a recall value aroung 0.75 but only a precision around 0.6.

		predicted				
		Class 1	Class 2	Class 3	Class 4	Class 5
real	Class 1	36.2	2.35	7.05	2.1	2.3
	Class 2	3.4	26.55	13.1	2.05	4.9
	Class 3	3.35	5.35	38.1	1.8	1.4
	Class 4	3.3	3.1	7.3	33.2	3.1
	Class 5	0.95	1.3	1.55	0,5	45,7
	$\sum$	47.2	38.65	67.1	39.65	57,4

Table 1: Confusion matrix for five class classification. The sum is the total number of images assigned to a class. The diagonal shows the correct predictions, all other matrix entries show the distribution of the false predictions.

The results for the classification of 15 classes is presented in Figure 6. Again, the five classes from the five class classification are included. A further deterioration of the precision and recall of the experiment can be noticed. Class 5 which performed best in the 5 class classification also gets good results in this classification. Other classes like class 9 and 10 show recall and precision values below 0.3. Classification results for this classes are not reliable. This causes the whole classification for 15 classes to be not as reliable as a classification with less classes. More work in this area is needed to obtain satisfying results.

It can be noticed that the classification result is highly dependent on the number of classes. The result deteriorates with the number of classes used. It was also striking that some classes seem to disturb the result extremely. If one of this classes was included in the experiment, the total result was noticeably worse. This is due to the high intra class variance of these classes. This causes the classifier to relax constraints for these classes which abets wrong predictions for all other classes.

7. Conclusion

In this paper a compact description of images was introduced. Using the *bag of keypoints* approach, the images were described by compact feature histograms. For the generation of the feature histograms, a tree-based feature space clustering was used. Based on this clustering, a codebook was generated. The tree-based feature space allows an efficient mapping of new features to the corresponding codebook entries. The compact features can be used for a multiclass object classification which was able to distinguish up to 15 different object classes. The classification performance is highly dependent on the number of classes used. For small number of classes, classification results are satisfying but for higher number of classes further work in this area is still needed. The inclusion of localization information and combination of classifiers could be a good way to receive better results.

References

[1] TUYTELAARS, T., MIKOLAJCZYK, K.: Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 2008, vol. 3, no. 3, p. 177–280

[2] LOWE, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004, vol. 60, no. 2, p. 91–110

[3] CSURKA, G., DANCE, C.R., FAN, L., WILLAMOWSKI, J. and BRAY, C.: Visual categorization with bags of keypoints. In: *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, p. 1–22

[4] HUISKES, M.J., LEW, M.S.: The MIR Flickr Retrieval Evaluation. In: *Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval (MIR'08)*. Vancouver, Canada, 2008

[5] MACQUEEN, J. B.: Some Methods for classification and analysis of multivariate oberservations. In: *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, vol. 1. p. 281–297

[6] AIZERMAN, A., BRAVERMAN, E.M. and ROZONER, L.I.: Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 1964, vol. 25, p. 821–837

[7] GRIFFIN, G., HOLUB, A. and PERONA, P.: The Caltech-256. Caltech Technical Report

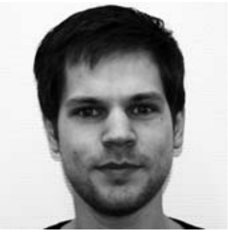
[8] VAPNIK, V., *The Nature of Statistical Learning Theory.* Springer New York, 1995

About Authors...

Iris HEISTERKLAUS

was born in Berlin, Germany in 1986. She received the Master's degree in Electrical Engineering, Information Technology and Computer Engineering from RWTH Aachen University, Aachen, Germany, in 2012 and is currently working as a Ph.D. student at the Institute of Communications Engineering, RWTH Aachen University. Her focus is on image and video analysis and content recognition.

Christopher BULLA



was born in Waldbröl, Germany in 1984. He received the Dipl.-Ing. degree in Computer Engineering from RWTH Aachen University, Aachen, Germany, in 2010 and is currently working as a Ph.D. student at the Institute of Communications Engineering, RWTH Aachen University focusing on image analysis, particular on local features and their use for Content based Image Retrieval.