Reducing Ambiguity of Local Descriptors for Visual Recognition

SOBHAN NADERI PARIZI

Master’s Thesis at NADA
Supervisor: Alireza Tavakkoli
Co-Supervisor: Ivan Laptev
Examiner: Stefan Carlsson

TRITA Sep 2010-21
Abstract

One of the most commonly used methods for representing visual information is to study properties of image regions at local neighborhood of pixels. Visual appearance, by its nature, is subject to extensive change under different conditions. Rotation, scaling, illumination, and viewpoint are examples of parameters which change appearance of objects in an image. Despite these changes, we as humans, are still amazingly good at detecting objects and perceiving their identity in an image. For computers, however, it is not easily achievable to tolerate such a great degree of invariance. The reason is that a human being sees visual elements in relation to each other and accordingly makes conclusions in a higher level than just local appearance of single image regions. For example, the fact that sky appears at the top of ground, vehicles move on roads, windows appear on vertical planes, cats are not likely to be seen in sea, etc.

Most of previous works has devoted their focus on detecting regions of image which are highly distinctive and has tried to transform visual descriptions in a standard form in order to cope with these visual variations. Nevertheless, there is much more to be studied regarding how to put these local features together in order to model their higher level dependencies more appropriately. Improved discriminability and expressive power of local features affects at early stages of visual recognition systems and propagates through the whole pipeline of the process.

In this thesis we try different approaches towards decreasing ambiguity of local feature representations by organizing local features based on some higher level information. We show that our way of decoupling/grouping local features results in consistent and significant improvements of performance of local features in several Computer Vision tasks.
## Contents

I Introduction to Computer Vision and Visual Recognition 3

1 Representing Visual Information (Image/Video Descriptors) 5
   1.1 Scale Invariant Feature Transform 7
   1.1.1 Keypoint Localization 8
   1.2 Histogram of Visual Words 9
   1.2.1 Limitations 11

2 Mathematical Models for Visual Recognition 13
   2.1 K-Nearest Neighbor Classifier 13
   2.2 Support Vector Machines 15
      2.2.1 Mathematical Formulation of SVMs 16
      2.2.2 Nonlinear Classification 17

II Disambiguating Local Features in Still Images 23

3 Spatial Aware Representations for Image Classification 27
   3.1 Introduction 27
   3.2 Context and Scene 29
      3.2.1 Fixed-Grid vs. OCG 30
   3.3 Our Approach 31
      3.3.1 Localizing The OCG 32
      3.3.2 Feature Extraction 33
      3.3.3 Classification 34
   3.4 Experimental Results 35
      3.4.1 Database 35
      3.4.2 Implementation Details 35
      3.4.3 OCG Configuration 35
   3.5 Conclusion and Future Works 39

4 Attributed Local Features for Image Classification 41
   4.1 Introduction 41
   4.2 Local Feature Encoding 44
      4.2.1 Spatial Grids 44
III Disambiguating Local Features in Videos 53

5  Spatiotemporal Aware Representations for Human Action Recognition 55
   5.1 Introduction ................................. 55
   5.2 Related Work .............................. 56
   5.3 Feature Types and Recognition Approach .... 58
       5.3.1 Feature Descriptors .................. 58
       5.3.2 Classification Approach ............. 58
   5.4 Experimental Results ..................... 59
       5.4.1 Database ............................. 59
       5.4.2 Annotation of Actor Bounding Box .... 59
       5.4.3 Evaluations .......................... 60
   5.5 Conclusion and Future Works ............. 61

6  Attributed Local Features for Human Action Recognition 63
   6.1 Introduction .............................. 63
   6.2 The BoF framework ....................... 64
       6.2.1 Features ............................. 64
       6.2.2 Histogram Representation .......... 64
       6.2.3 Multi-channel SVM ................... 65
   6.3 Improving BoF with Non-local Cues ....... 65
       6.3.1 Motivation ........................... 65
       6.3.2 Enriching BoF Representation with Region-level Information 66
       6.3.3 Spatio-temporal Grids ............... 67
       6.3.4 Foreground/background Motion Segmentation .... 67
       6.3.5 Action detection ..................... 68
       6.3.6 Person Detection ..................... 69
       6.3.7 Object Detection ..................... 70
   6.4 Experimental Results ..................... 70
       6.4.1 Baseline Performance ............... 70
       6.4.2 Improvements with Channel Combination .... 71
Introduction

Computer vision has recently got widespread in different fields of science and technology. Computer graphics, virtual reality, cinema technology, surveillance and security, medical imaging, internet, robotics, and computer games are all examples where computer vision methods are widely used in practice. The growing interest in working with photos and videos in addition to the great deal of data available (raw and even annotated) as well as increasing number of its applications has made computer vision field one of the most appealing and hot topics for research. In this thesis we intend to introduce some of the basic applications of computer vision, discuss its standard algorithms, and propose some extensions to some of the currently available methods in the field.

Information extracted from a local neighborhood around a certain point in an image is referred to as local descriptor. Aggregating local descriptors extracted from multiple locations of an image yields a representation which is analogous to basic algorithms used in text retrieval methods such as frequency histogram of keywords in a text document. Such like methods, as will be seen later in this thesis, has shown impressive performance in a wide variety of computer vision tasks.

Local image descriptors are essential building blocks in the majority of current approaches to object and scene recognition [22, 38, 82], image retrieval [69, 60, 8], image segmentation [33], 3D reconstruction [27] and image restoration [45]. The success of local features can be attributed to the high discriminative power and stability of modern local descriptors playing a crucial role in all of these above applications. Much of the thorough work has been devoted in the past to the design, evaluation and optimization of local image descriptors [43, 48, 49, 79].

Local descriptors should balance a trade-off between the discriminative power and invariance needed to overcome image variations due to projective effects, lighting variations, individual variations within a class and other factors. The need of invariance usually implies a limit on the discriminative power of individual local features. To overcome this limitation, previous work usually aggregates information from features at different locations of an image for example by constructing bag-of-features representations for image classification and retrieval [38, 82, 60], by designing spatial feature constellation models for object recognition [22, 20] or by using geometric verification for image retrieval and 3D reconstruction [27, 60].

In this thesis we try to improve discriminative power of local descriptors by augmenting them with extra information which goes beyond their local extent. We
do this either by adding global spatial constraints or by injecting semantic level information to local descriptors.

The rest of this thesis is organized as follows. The thesis is divided into three parts. Part I provides a brief introduction to the field of computer vision. This part introduces basic concepts, key terms, and alternative approaches relevant to our line of work. The concepts which are used throughout the rest of the thesis has been discussed in more details. In particular, chapter 1 discusses how visual content is represented in a computer and chapter 2 discusses alternative strategies for modeling and recognizing visual patterns. In Part II we propose some ideas for disambiguating local descriptors in still images. We validate our ideas by performing experimental evaluations on different tasks and different databases. In particular, in chapter 3 we study effect of spatial information of objects on expressive power of histogram representation of local features. In chapter 4 we propose a general idea to reducing ambiguity of local image features by adding higher level attributes to them. Finally, in Part III we study ambiguity of local features in video classification problem. Analogous to the previous part, in chapter 5 we study how inclusion of spatial and temporal information of local features affects recognition of realistic human actions. Again, similar to the previous part, we propose to add higher level attributes to local spatiotemporal features. In chapter 6 we demonstrate how these global attributes help us to disambiguate between histogram of local descriptors. We conclude the discussed problems in chapter 7.
Part I

Introduction to Computer Vision and Visual Recognition
Chapter 1

Representing Visual Information
(Image/Video Descriptors)

Each gray-scale image can be seen as one realization of a two dimensional real-valued signal. However, in order to the representation mathematically more tractable two levels of discretizations are applied. In the first level the two dimensional signal is sampled in different points within the image frame. These discrete points are referred to as pixels. Up to this point the image is represented by a two dimensional matrix of real values. The second level of discretization is to quantize value of each pixel up to a certain resolution (e.g. 256 different values). This is what we refer to as digital image. From now on throughout the text by image we mean digital image.

Representing images as gray value of their pixels, as they are, does not provide a powerful and discriminative representation unless the images are fairly accurately registered and properly normalized to cope with different variations including illumination, scale, etc. A more appropriate way of describing content of images is to represent images as a measure of how they respond to different local filters. Inspired by response of biological visual systems in mammals, a set commonly used filters is sinusoidal filters in different orientations and frequencies convolved with Gaussian functions with different variances. Filters of this types are usually referred to as Gabor filters [24]. Similar to Gabor filters, we have LM filters introduced by Leung and Malik [40], S filters introduced by Schmid [63], and MR filters introduced by Varma et al. [75]. These three sets of filters (filter banks) are illustrated in Figure 1.1. LM filters include first and second order derivatives of Gaussian and oriented Gaussian filters in several orientations and scales which can be seen as local detectors for edges, bars, and spots in the image. S filters are a set of rotationally invariant Gabor-like filters. MR filters are comprised of 14 different filter masks whereas only 4 responses are returned as the descriptor. This is done by maximizing response of masks at different orientations which results in rotation invariancy.

An alternative approach to using response of filter banks is to use gray value of pixels but in a local neighborhood around each pixel e.g. 5x5 or 7x7 patches centered
CHAPTER 1. REPRESENTING VISUAL INFORMATION (IMAGE/VIDEO DESCRIPTORS)

(a) Leung Malik (LM) filters

(b) Shmid (S) filters

(c) Maximum Response (MR4) filters

Figure 1.1: Three different sets of filter banks are illustrated. Each filter bank comprises of a set of filter masks. By convolving each mask with input image at a certain location (pixel) we get the filter response which can be used as a descriptor for that specific pixel. After applying all of the masks to a certain pixel in the image, that pixel can be described by the vector of response values from different masks.

at a pixel [76]. We can then describe each pixel in the image by vectorizing gray-scale values in this local patch (e.g. matrix of 5x5 or 7x7 values).

These two types of features that we have discussed so far, i.e. vector of filter responses and gray-scale pixel values at a local neighborhood, are widely used to model visual properties of texture of a surface. Accordingly, their performance has been shown to be promising in many applications including material recognition [75, 40, 70, 76], object/image classification [80, 63, 64], and Object [64], text [59], and vegetation segmentation [57].

In addition to these, Local Binary Patterns [55] have also been shown to be useful for describing texture properties of image regions [61, 9, 31]. LBP works based on relative gray-scale value of neighboring pixels and therefore, inherently, is invariant to illumination.

Typically, texture does not contain much information about shape of objects although it best describes appearance of the material making up the object. For example surface of a certain type of wood is different from surface of soil or concrete...
1.1 Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) [43] is probably the most commonly used method to extract robust and discriminative local features in still images. The main idea of SIFT is twofold. First, it finds a set of highly distinctive local regions (referred to as keypoints) in an image. Second, it extracts features/descriptors which are invariant to scale and orientation changes. They have also shown to be robust to illumination and viewpoint change. Putting these two stages together, i.e.

in terms of their texture properties. However, texture analysis will not probably tell us much about the whole structure of the wooden object though, i.e. whether it is a wooden chair or a wooden wall. Nevertheless, texture is a useful measure to distinguish between a concrete wall from a wooden wall.

Structural patterns, on the other hand, can be better described by looking at dominant orientation of edge elements in the image. Figure 1.2 illustrates a representation of structural properties using oriented gradients for two objects i.e. Person and Bicycle. As shown in the figure, outline of structure of the object can be seen in this representation. The sketch shown in Figure 1.2 is based on so called HOG (Histogram of Oriented Gradients) representation introduced by Dalal and Triggs [13].

An alternative approach to HOG for capturing structural information is SIFT (Scale Invariant Feature Transform) [43] which is probably the most commonly used feature representation during the last half a decade. As we will see in the next chapters, we also use SIFT features extensively in many of our experiments throughout this work. Therefore, in the next section we explain SIFT features and discuss its applications in more details.

Figure 1.2

(a) Person
(b) Bicycle
CHAPTER 1. REPRESENTING VISUAL INFORMATION (IMAGE/VIDEO DESCRIPTORS)

Figure 1.3: Scale-space over which keypoints are detected. The three planes shown in the figure contain $D$ values (as in Equation 1.1) for three consecutive scales (in each plane the scale parameter $\sigma$ is larger from its next plane by a constant factor ($\sqrt{2}$)). If $D$ value at the pixel shown by the cross is an extremum (either minimum or maximum) w.r.t all of its 26 neighbors (shown by circles) then location of the pixel containing the cross as well as its corresponding scale level is detected as one distinctive keypoint.

Keypoint localization and region description, SIFT idea has shown to be successful in many Computer Vision tasks including object/image matching [60, 68, 49], object recognition [53], image classification [6, 46], and scene recognition [38, 5]. Moreover, SIFT is computationally light so that it can be used in realtime robotics applications such as localization and mapping as well [66].

1.1.1 Keypoint Localization

SIFT detects regions in an image which have strong edge elements and are highly distinctive in their local neighborhood. Edge response can be formulated as Laplacian of Gaussian which can be approximated from difference of Gaussian filters with different smoothing parameters ($\sigma$ of the Gaussian kernels):

$$D(x, y, \sigma) = G(x, y, \sigma_1) * I(x, y) - G(x, y, \sigma_2) * I(x, y)$$  \hspace{1cm} (1.1)

In Equation 1.1, $I(x, y)$ denotes gray-scale value of the pixel at $(x, y)$ location of the image. $G$ is Gaussian kernel and $*$ denotes convolution operator. In order to find locations whose $D$ value is extremum, each location is compared to its 26 neighbors as shown in Figure 1.3. If it is larger/smaller than all of its neighbors then that location in addition to its scale level is returned as one keypoint.

This procedure results in detection of a few thousands of keypoints in a 500x500 image. This means that only around 0.8% of pixels of the image are detected as keypoint.
1.2 Histogram of Visual Words

So far we have explained a few examples of different ways of describing image content which then leverages mathematical derivations and computational processes on an image. In order to do recognition in an image, however, we need to put local features extracted from different locations in the image altogether so that we could have a global representation of the image. This global representation would finally depend on the framework that we choose for recognition though.

One of the simplest, and yet most popular, ways of representing local features is the Bag of Features (BoF) paradigm. Therefore, in the this section, we explain BoF framework in more details.

Inspired by document classification and text mining methods, one of the mostly used strategies for capturing visual information is to count frequency of some known elements (as words in a document) in an image. Such a strategy is widely used for vision tasks such as classification in still images [64] as well as in videos [36].

Assume a document classification task where given a text document (e.g. an email) the problem is to discover category of the document among some predefined classes (e.g. spam and non-spam). In image classification problem, analogous to this toy example, we are given an image and the problem is to discover whether there exists an object (e.g. car) in the image or not. In the document classification example it is sensibly reasonable to count frequency of occurrence of some keywords in it, as building blocks of a document, and to model distribution of those keywords by a histogram. Now the question is what are building blocks for an image? It turns out that we do not have an obvious interpretation for what images are comprised of.

We can, however, make a virtual universal interpretation in order to represent building blocks of an image. One possible strategy is to consider images to be comprised of a collection of small patches each of which represents a distinct visual element. As a toy example, one could consider image of a human face to be comprised of small patches containing the following elements: eye, eyebrow, lip, nose, ear, etc. (Figure 1.4). However, in practice, the visual building blocks are not necessarily this much interpretable. Instead, they are represented by local descriptors of image regions (e.g. SIFT) as we discussed earlier in this chapter. Such visual elements are referred to as textons. In other words textons has more or less the same role in an image that a word has in a document. Accordingly, analogous to word dictionary in a text domain we can define a visual vocabulary (also called bag of visual words) in image domain.

Now we are ready to take the final step towards the concept of BoFs in vision.

---

1. This interpretation, as we will see later, is subjective in the sense that for each problem in hand the set of elements (building blocks) which compose an image depends on the type of the dataset we are working with; e.g. if we are working only with faces then eye, nose, ear, etc. can be considered as the aforementioned elements while if we are working with database of different flowers then small image regions with different colors or different curved shapes might be considered as the elements.
community. In BoF framework, we take a random subset of local features from a set of training images to create a visual vocabulary. Once the visual vocabulary is built we can represent any given image by building frequency histogram over the elements of visual vocabulary. This can be done by counting the number of times a visual word (a certain element of the visual vocabulary) is seen in the image. It should be noted that each visual word is now represented by a vector of feature values. On the other hand, the set of local features that we extract from an image are also a vector of feature values (e.g. SIFT). It is highly unlikely that we can find exact duplicates of visual words among local features of the image. Number of distinct local features is infinite while size of visual vocabulary that we build is definite and usually smaller than 10000\(^2\). Therefore, we need to define a similarity measure between a given visual vocabulary and a local feature vector extracted from an image. Using the similarity measure we can assign each local feature to an element of our visual vocabulary. This scenario is shown in Figure 1.5. Figure 1.5b shows the image obtained by replacing each pixel of the original image (Figure 1.5a) with the label of the visual vocabulary element to which the local feature extracted from that pixel is assigned to. Therefore each pixel in Figure 1.5b takes a scalar value between 1 an K where K is size of visual vocabulary (50 in our example). Such an image is called texton map.

In order to show that visual words can be good representatives of local feature vectors we have tried to transit from Figure 1.5b back to the original image. Result of this transition is shown in Figure 1.5c. Such a transition is not always possible, but, because of the type of local features that we have used in this example, being actual pixel values of a 5x5 local patch around each pixel, this can be done. As

\[^2\text{There are applications where size of visual vocabularies that are used is in the range of million as well.}\]
1.2. HISTOGRAM OF VISUAL WORDS

Figure 1.5: Original image (Figure 1.5a), texton map of the original image obtained by replacing each pixel in the original image by index of the element of visual vocabulary which is closest to the local feature vector extracted from that pixel (Figure 1.5b), and the result of an attempt to reconstruct the original image from its texton map (Figure 1.5c).

Figure 1.5c shows, some details are lost in the reconstructed image whereas the main content of the image is still preserved.

1.2.1 Limitations

Despite their simplicity and expressive power, local descriptors are ambiguous which is basically due to their locality. This issue can be compensated for to some extent by devising techniques to incorporate global information into BoF representation, and more generally, into local descriptors. Within Chapter II and III we will study the problem of ambiguity of local descriptors and will propose some solutions for it.
Chapter 2

Mathematical Models for Visual Recognition

In the previous section we discussed some ways of capturing information from an image and representing it in a mathematically and computationally sound form. In this section we briefly go through some of the most commonly used strategies by which we can do learning as well as recognition. In the learning part, we are given a bunch of image features coming from different samples of images. Our task is then to build a mathematical model by which we can best describe the given samples. These learned mathematical models may also be used to classify images between two (or more) classes of concepts or to percept their common underlying pattern. In recognition we already have the mathematical model learnt. Instead, given an input (test) image, we have to examine the image with the model and report the level of consistency of that image with our model.

Following in this chapter we review two common classification paradigms i.e. \textit{K-Nearest Neighbor} methods and \textit{Support Vector Machine}, a.k.a. SVM.

2.1 K-Nearest Neighbor Classifier

In a classification problem we are given a set of training samples in the form of \( D = \{(X_i, y_i)\}, i = 1..N \). In this notation \( D \) is the training set, each element of this set, i.e. \((X_i, y_i)\) pair, is one training sample where \( X_i \) denotes vector of feature values and \( y_i \) is label of that feature which denotes the category that \( X_i \) belongs to. And \( N \) is number of training samples. Given the training set \( D \) and a test sample \( X_{test} \) goal of a classification problem is to find the corresponding class where \( X_{test} \) belongs to, i.e. to find \( y_{test} \).

A simple solution to a classification problem is to find the training sample nearest to the given test sample and copy class label of that sample as \( y_{test} \). This solution, despite its simplicity, works fairly good for many applications. Performance of this method, however, is very dependent on distribution of training samples and small displacements or few removal/addition of training samples could have remarkable
influence on decision boundary of the classifier and accordingly its performance. Therefore, this method, as shown in Figure 2.1, is susceptible to overfitting to training data. This method is referred to as 1st.-Nearest Neighbor also called 1-NN. In its general form, we have K-NN where the first $K$ nearest neighbors of the $X_{test}$ are found and then the class label with the majority of occurrences is assigned to $y_{test}$. Setting $K$ to a value bigger than 1 helps to avoid overfitting issue.

Figure 2.2 illustrates how changing value of the parameter $K$ effects decision boundary of a K-NN classifier in a toy example. As it is shown, $K = 1$ perfectly fits to training data with an arbitrary complex decision boundary. As we increase value of $K$ to a certain extent the decision boundary gets more and more smoothed and generalized.

Despite the fact that values bigger than 1 could be more appropriate for a given task, practically, it is not always possible to increase value of $K$ arbitrarily. Time complexity of the problem of finding the first $K$ nearest training samples to a given $X_{test}$ is highly dependent on the value of $K$ as well as number of training samples, $N$. So, as we increase value of $K$, we might get to the situation that K-NN is no longer computationally feasible. Nevertheless, for the case of 1-NN there are fast algorithms such as $k$-d tree algorithm [25] as well as yet faster approximations such as BFF algorithm [4]; therefore, 1-NN is a commonly used version of K-NN.
2.2. SUPPORT VECTOR MACHINES

Figure 2.2: We want to use K-NN to solve classification problem for a given training sample of a two class problem (similar to Figure 2.1). Figures 2.2a-2.2d show effect of changing value of parameter $K$.

2.2 Support Vector Machines

Support Vector Machines (SVMs) has recently gained great popularity in the field of Computer Vision due to their impressive classification power. SVM, however, in its mostly used and general form, was introduced in 1995 by Cortes and Vapnik [11]. Although, they used SVMs for recognition of isolated handwritten digits, but, nowadays Support Vector Machines are showed to be powerful enough to be expandable to much more complex problems. Given a set of labeled samples from a two class problem, SVM finds the optimal separating hyperplane in an extremely high dimensional augmented feature space. Optimality of the separating hyperplane is measured by the size of margin between decision boundary and sample points. This leads us to a deterministic solution provided that the sample points are linearly separable [7]. After applying slight changes, however, the idea can also be extended to the case where samples are not perfectly linearly separable [11].
CHAPTER 2. MATHEMATICAL MODELS FOR VISUAL RECOGNITION

(a) Linearly separable two-class problem
(b) Arbitrary separating hyperplanes (lines)
(c) Max-margin separating hyperplane

Figure 2.3: Toy example for a linearly separable two-class problem. 2.3a shows sample points. 2.3b shows two arbitrary decision boundaries which classify the samples correctly. 2.3c shows the decision boundary with maximum margin. Crossed sample points are the ones which restrict margin of the classifier (so called Support Vectors).

2.2.1 Mathematical Formulation of SVMs

In this section we assume that we are given a two-class classification problem (unless otherwise stated). Figure 2.3 illustrates a toy example of a set of linearly separable samples in 2D. This is a two-class problem and the goal is to separate red samples from the blue ones. As it is shown in Figure 2.3b there are infinitely many lines which correctly separate the samples. The question is which one to choose! As we will see later, it is highly preferable to choose the separating line which maximizes the margin between decision boundary and sample points of the two different classes [7]. Such a separating line is demonstrated in Figure 2.3c.

Complexity of a family of classifiers is represented in terms of their expressive power in modeling arbitrary patterns with different levels of complexity. The notion of complexity is sometimes referred to as capacity of classifier as well. For example capacity of high order polynomials is larger than straight lines; or equivalently, high order polynomials are more complex than straight lines. This means that if you choose to classify a set of samples using a high order polynomial as your classifier, you can train your classifier in such a way that the samples are correctly classified for some cases where no straight line can classify the samples. However, it is not always good to use complex models. As stated in the law of succinctness:

"Entities should not be multiplied unnecessarily". Occam’s Razor

The reason is that the more complex you make a rule the less likely it would be that your rule fails to generalize to a bigger set of samples. Figure 2.4 illustrates a two-class problem and a sequence of classifiers which tend to become more and more complex. It is obvious that the samples are not linearly separable (in their original feature space). Figure 2.4a represents an exemplar linear classifier which, of course, is subject to some misclassifications. Figures 2.4b-2.4d illustrates more complex classifiers. Even though all of the samples are correctly classified in 2.4d, however, it is highly likely that this complex decision policy does not hold for many
newly seen samples; a few examples of such samples are denoted in the figure by crosses. Color of the crosses shows the class to which they belong, however, they are misclassified by the decision boundaries of Figure 2.4d. Therefore, using a complex pattern which perfectly fits to our training data (assuming that the data points hold a sophisticated distribution) is opposed to generalization power of classifier, and therefore, an optimal classifier has to tradeoff between these two criteria i.e. classification accuracy and generalization power. The decision boundary which maximizes the classification margin (shown in Figure 2.3c) handles this tradeoff automatically.

Assume that we are given a set of training data
\[ D = \{(x_i, y_i), i = 1..N\} \]  
(2.1)

where \( N \) denotes number of data samples, \( x_i \) denotes feature vector for \( i^{th} \) sample, and \( y_i \) is its class label. Recall that we are working with a two-class problem, and therefore, \( y \in \{-1, +1\} \).

In fact, we are looking for a separating hyperplane in the following form:
\[ w \cdot x + b = 0 \]  
(2.2)

Here, vector \( w \) represents our separating hyperplane. However, in order to include the notion of max-margin into our equation we write two important properties of such a hyperplane:
\[ w \cdot x_i + b \geq +1, \forall i|y_i = +1 \]
\[ w \cdot x_i + b \leq -1, \forall i|y_i = -1 \]  
(2.3)

Or equally:
\[ y_i(w \cdot x_i + b) \geq +1, \forall i \]  
(2.4)

It can be shown that size of margin of the classifier can be calculated from \( w \) in the following way [11]:
\[ \text{sizeofmargin} = \rho = \frac{2}{||w||} = \frac{2}{\sqrt{w \cdot w}} \]  
(2.5)

Goal of SVM, therefore, is to maximize \( \rho \) under the constraints stated in 2.4.

2.2.2 Nonlinear Classification

So far, we have assumed that the sample points are linearly separable. What if there is no linear hyperplane which can perfectly separate the sample points? In fact the latter is almost always the case in practical problems i.e. decision boundary usually is nonlinear. There are two solutions to this problem. The first solution to is modify Equation 2.4 to handle the samples which will inevitably be misclassified.
Figure 2.4: Decision boundary generated by SVM for a toy example for different settings of parameters. 2.4a shows decision boundary of a simple linear classifier. 2.4b shows an appropriately trained classifier. 2.4c shows a case where the model is over-fitted to training data. 2.4d represents an extreme case when over-fitting happens. Crosses show exemplar unseen data points which would be misclassified by this over-fitted classifier.
by the linear classifier. This involves adding one new term to the equation which measures degree of misclassification for each sample. This term is usually referred to as slack variable and is denoted by $\xi$. As a result we get

$$y_i(w.x + b) >= +1 - \xi_i, \forall i$$ (2.6)

Also in the optimization part we penalize the hyperplane according to its degree of misclassification for different samples. Therefore, our optimization problem is:

$$\min_{w,\xi} \{||w|| + C \sum_{i=1}^{n} \xi_i\}$$ (2.7)

Here $C$ is a constant which moderates penalty of misclassification. This value of this constant is usually tuned by cross-validation.

The second alternative solution for the cases where sample points are not linearly separable is to expand the data into a higher dimensional space using nonlinear transformations denoted by $\varphi$. So far we have been trying to classify samples in the same space as the features are defined. The idea is to use transformations which take us to a more complicated space than the feature space (probably with many more dimensions). Therefore, $\varphi$ is a function from feature space $\mathbb{R}^n$ to a higher dimensional space $\mathbb{R}^N$ where $n$ is the length of feature vector and $N > n$ i.e. $\varphi : \mathbb{R}^n \mapsto \mathbb{R}^N$. As a simple example of kernel functions we can refer to polynomial kernels. A polynomial kernel of degree $d = 2$ maps feature vector of the form $x = [x_1, x_2]$ to a new vector $\varphi(x) = [x_1^2, x_2^2, x_1.x_2]$ and if you append 1 to $x$ before transforming it by $\varphi$ you could get a constant term and the original vector $x$ in the transformed spaces as well i.e. $\varphi(x') = [x_1^2, x_2^2, x_1.x_2, x_1, x_2, 1]$ where $x' = [x_1, x_2, 1]$. Now it is easy to see that a linear classifier on the transformed space, $\varphi$, is a nonlinear classifier in the input feature space, $x$. Figure 2.5 illustrates a toy example of a two-class classification problem where no linear solution can help (Figure 2.5a. Nevertheless, a nonlinear classifier (Figure 2.5c) can perfectly classify the samples. Now, if we transform the sample points to a higher dimension by adding squared sum of 2D coordinates of each points as its third dimension (Figure 2.5b), samples can be linearly separated by a plane in 3D space (Figure 2.5d).

However, when degree of the polynomial is chosen to be large and our input features are too long (which is the case in most of practical problems) this transformation $\varphi$ becomes massively multidimensional. This makes our problem computationally intractable.

Kernel trick helps us to overcome this computational intractability of transformed features. If we continue with mathematical operations involved in SVM we will see that it only involves inner products of feature vectors. It can be shown that $\varphi(x).\varphi(y) = \varphi(x,y)$. Therefore, we can avoid computations in the transformed space. Instead, we can calculate dot product of feature vectors in the feature space and then apply the transformation on their result which is essentially a scalar value. This can also be written in the form of kernel functions as follows. A kernel function is a function that maps each feature space to a real value scalar, $K : \mathbb{R}^n \mapsto \mathbb{R}$. In
Figure 2.5: Extension of sample points to spaces with higher dimensions can make the classification problem simpler. 2.5a shows some sample points in 2D space representing two different classes. Obviously, there is no linear classifier in this 2D space which can correctly classify the sample points. 2.5a shows the same sample points when a third dimension is added to the points which is sum of squared values of 2D coordinates of the points. 2.5c shows a non-linear decision boundary in the 2D space. 2.5d shows a linear classifier in 3D which can correctly classify all of the sample points.
2.2. SUPPORT VECTOR MACHINES

In other words, a kernel function gets a pair of samples from feature space, applies some computation on them in a higher dimensional space, and returns back the real value scalar result of the computation. Some of the commonly used kernels are:

- **Linear:** $K(X_i, X_j) = \varphi(X_i) \cdot \varphi(X_j) = X_i \cdot X_j$
- **Polynomial:** $K(X_i, X_j) = \varphi(X_i) \cdot \varphi(X_j) = (X_i \cdot X_j + 1)^d$, $d$ denotes degree of the polynomial
- **RBF:** $K(X_i, X_j) = \varphi(X_i) \cdot \varphi(X_j) = \exp(-\gamma \|X_i - X_j\|^2)$, $\gamma > 0$

Here $X_i, X_j$ are two feature vectors.
Part II

Disambiguating Local Features in Still Images
Despite the fact that BoF scheme yields a simple and yet expressive representation of images, it does not encode all information of an image. For example, BoF results in an orderless representation, and therefore, spatial information of local features are completely ignored. More importantly, global information are also not captured in BoF encoding because of their local nature. To name some types of these global information we can refer to the global structure of natural scenes; e.g. sky is naturally on the top of ground, cars typically appear on road, presence of a person with a tennis-rocket in his/her hand and a tennis-ball in its close vicinity increases likelihood of having tennis-court in the image as well as happening of play-tennis action. As yet another example, Figure 2.6 demonstrates the level of importance of spatial configuration in visual recognition. Obviously, Figure 2.6b is a much more convenient representation of a face even though it has exactly the same set of local descriptors as that of Figure 2.6a.

One could count many more informative samples of cases where informative cues are left untapped in BoF representation.

Being able to encode global information and spatial constraints of local features within BoF paradigm helps to increase its discriminative power, and accordingly, its recognition performance. Furthermore, sometimes it happens that local features extracted form irrelevant locations/regions of an image are labeled to the same visual vocabulary element in BoF paradigm. This type of ambiguity usually happens because of loss of precision in the quantization (labeling) of local features while building BoF histograms. For example, if we have means to segment regions such as side-walk, road, and parking-area in an image (denoted by $R1$, $R2$, and $R3$ in Figure 2.7 respectively) we can use this information to discriminate between features of the following three types of actions respectively: run/walk, drive-car/ride-motorbike, and get-out-car/open-trunk even if some features are ambiguous.

In this part we propose some methods to decrease this ambiguities. More specifically, in Chapter 3 we make an endeavor towards incorporating spatial information.
Figure 2.7: Segmentation of regions of an image into *sidewalk* (denoted by $R1$), *road* (denoted by $R2$), and *parking-area* (denoted by $R3$). These regions correspond to actions such as *walk*, *drive-car*, and *get-out-car* respectively. Such like segmentation can be helpful in disambiguating similar features extracted from these regions, and therefore, help to discriminate between the three aforementioned types of action classes.

and holistic structure of images into BoF representations. There we will take a perspective towards modeling image context and will try to devise spatial information into BoF representation so that it helps to model context of objects more effectively.

In Chapter 4 we propose a general framework for injecting global information into local features in order to disambiguate local features. As we will explain later in more details this is done by augmenting local descriptors with some labels coming from semantically coherent regions in the image.
Chapter 3

Spatial Aware Representations for Image Classification

Context plays valuable role in any image understanding task and numerous studies have shown the importance of contextual information in computer vision tasks, like object detection, scene classification and image retrieval. Experiments on human perception on task of scene classification and visual search have shown that human visual system makes extensive use of contextual information as post-processing in order to index objects. Several recent computer vision approaches use contextual information to improve object recognition performance. They mainly use global information of whole image by dividing the image into several sub regions, so called fixed-grid. In this chapter we propose a new approach to retrieval contextual information, by localizing the center of these grids based on salient objects of each image. We claim this approach will form contextual features in a much more coherent and informative way than fixed-grid strategy. To compare our results with the most relevant and recent papers, we use Pascal 2007 data set. Our experimental results show a superior improvement in terms of Mean Average Precision.

3.1 Introduction

Contextual representation of natural images has recently turned into an interesting field of study in computer vision. Torralba uses contextual information to predict location and size of different classes of objects in an image [72]. Once this prior information is captured, one can run object detectors only on the promising locations of the image and make the detection process faster by saving a lot of computation [51]. In [14] context is used in conjunction with intrinsic object detectors to increase performance of the detector and enhance position and size of the detected bounding boxes. Marszalek et al. [47] have shown that many of the human actions in real movies can be classified better in relation to their visual context. Another field of study where context plays the most critical role is classification of scene categories. In [5, 38] it has been shown that different images of an individual scene category
CHAPTER 3. SPATIAL AWARE REPRESENTATIONS FOR IMAGE CLASSIFICATION

share a lot of information in their global representation.

People take different approaches to exploit the contextual information of images. Sometimes, the image background at the local surroundings of the object is used as the context of the object [32]. Similarly, it has been shown that surroundings of object bounding box contain informative support for classification of animals [44]. Oliva et al. [56] models the context with a holistic representation over the whole image. Some others [29, 62] model the context by using the relation between objects and estimate the probability of co-occurrence of different objects and their correspondence constraints.

Many of the computer vision methods make benefit of histograms today [36]. Simplicity of the histogram-based methods associated with their effectiveness in a wide variety of problems has made them very popular and vastly used. One of the most general implementations of histogram-based methods is referred to as Bag of Features (BoF) [64].

As mentioned before, despite the enormous use of histogram-based methods, in general, they all suffer from an inherent drawback being lack of spatial information. Several papers have been published trying to incorporate spatial information in the BoF framework. Lazebnik et al. introduces spatial pyramid idea where the image is subdivided into smaller regions several times and BoF for each region is calculated [38].

Marszałek et al. [46] create a fixed-grid over the whole as shown in Figure 3.1. For each grid cell, one BoF is calculated independently. Then these BoFs are concatenated together to form the final feature vector. This method, in general, does not store the spatial information explicitly. The intuition behind this idea is that each individual BoF models distribution of each spatial region of the image and therefore the spatial information is preserved. Some others explicitly add the x, y coordinates of each cell to the end of feature vector in order to retain the spatial information instead [5].

The limitation of the grid-based methods is that they define a fixed grid over the image. Keeping the grid fixed according to the image frame causes the method to be variant to image translation. However, the fixed-grid idea can model typical scene layouts in a fairly efficient way. As we will show later in section 3.4, it performs much better than having no grid at all.

We suggest a new configuration for placing the grid on image so that the spatial information is preserved and besides the extracted feature is robust to image translation and scaling. The idea is to adjust the grid according to location and size of objects in image. As the way of example consider the category of images containing a scene of sea in the scene classification problem. If we try to adjust the grid such that the center cell always lies over the sea, we can most likely expect to have sky the top cells and beach (or land) at the bottom cells.

Basically, we modify size and position of the fixed-grid such that the center cell of the grid fits to the bounding box of one or more objects which are present in the image. Thus, we call the proposed method Object Centered Grid or briefly OCG.

Our main contribution in this chapter is, therefore, to show that the feature
3.2. CONTEXT AND SCENE

Figure 3.1: One smart idea for incorporating spatial information in the BoF framework is to make a fixed-grid on the top of the image. An independent BoF is computed for each grid cell and then all of the BoFs are concatenated together in a fixed order. According to the image, the cells at the upper row are expected to cover sky and roof of the buildings; the middle cells are expected to cover a mixture of road, sidewalk and buildings; and the bottom cells are expected to mainly cover the road.

vector extracted from OCG represents scenes better than fixed grid counterpart. We claim that OCG forms features in a much more coherent and informative way than the fixed-grid strategy. Our final goal here is not to beat the state-of-the-art method but to measure the maximum extent of gain in terms of average precision that one could achieve by using OCG comparing to the fixed-grid approach.

The rest of the chapter is organized as follows: in the next section we will briefly address concept of context in computer vision as well as basics of our OCG idea. In section 3.3 we will explain the OCG method in more details. In particular, core of OCG method which is how to grid the image is discussed in 3.3.1. Section 3.4 contains evaluations of the OCG method. We conclude this chapter in section 3.5.

3.2 Context and Scene

Torralba specifies that context is beneficial in object recognition tasks, in particular, when intrinsic object features are not clear enough [72]. In other words, for images where the target object can be detected accurately enough by means of local detectors, it is preferable to rely on the result of the detector.

Contextual information comes into play when the object is locally hard to be detected. For example, street context is a strong clue to persuade existence of cars within the image. Even if the car is highly occluded or, for whatever reason, local information fails to single the car out, contextual information can still vote for existence of the car.
3.2.1 Fixed-Grid vs. OCG

The main advantage of using fixed-grid scenario in building histograms is to incorporate spatial information into BoF framework. In the fixed-grid framework, the image is divided into different spatial segments by putting a grid on it (Figure 3.1). Each cell models the part of the image that lies underneath it independently from the other cells. Afterwards, the BoF histograms calculated for the grid cells are concatenated into a bigger histogram. This concatenated BoF is the final feature vector that basically classifies the test image into one of the target categories. Spatial information is implicitly stored in the fixed-grid framework being the fact that the first part of the final BoF always models the upper left region of the image; the second part of the final BoF always models the upper middle region of the image, and so forth. Finally, the last part of the BoF models the lower right region of the image.

Here is the question; what if the image is translated such that the image does not contain much of the sky (Figure 3.2a)? On the contrary, what if it is translated in a way that the image does not contain much of the road (Figure 3.2c)?

In the samples shown in Figure 3.2, it is no longer true that the BoFs attained from the upper cells represent the sky or buildings’ roofs (Figure 3.2a). Similarly, in Figure 3.2c, the middle-row cells represent sky and the buildings’ roof while we ideally expect to observe those regions in the topmost cells (cp. Figure 3.1). The same contradiction holds for the bottom cells of the Figure 3.2c.

In general, the fixed-grid idea works best when the images are fairly accurately aligned. In other words, if the goal is to classify different scenes, image alignment would mean to have the sea, buildings, dining table in the middle of the image for the sea scene class, street scene class, and kitchen scene class respectively.

The OCG idea suggests modifying the grid according to the objects within the image instead of expecting the images to be already aligned. Figure 3.2b, 3.2d show the OCG configuration while assuming the central object to be the car. Obviously, if the image is translated to any side, the grid cells will still represent almost the same region of the image in term of visual appearance. It means that if we use OCG approach, regardless of the image content, it is always expected to see road in the bottom cells, sidewalk at the middle cells and buildings and sky at the topmost cells.

Of course, one cannot expect the cells of an object-centered grid to always have a clean and unique content. Most of the objects in real world, e.g. car or person, are more or less context free [73]. You may see a person almost everywhere in any context. Therefore, surrounding of the object will not stay unique either.

In fact, our claim is that OCG produces much more coherent feature representation of image context than a fixed-grid representation; since it is localized according to a real object. For example, it is not likely that the bottom cells of an OCG contains buildings when we have a car in the center cell of the OCG.
3.3. OUR APPROACH

Our main goal in this chapter is to measure the amount of information that one could gain by incorporating the spatial information into the BoF framework through OCG paradigm. According to our final results, the information gain of image context is too much affected by the way that the context is modeled. Finally, we show that OCG forms the features in a very coherent way such that contextual information of the image holds a fairly stable correspondence with respect to the central object. Unlike OCG, formation of the contextual features in the fixed-grid representation is more of a random nature.

Our evaluations are mainly on image classification problems. It means that we have to classify images into a set of object category classes (e.g., car, dining table, boat). For each pair of test image and object category we have to say whether the image contains any instance of that object or not; regardless of its position in the image. The interesting point with the image classification problem is that objects play the most crucial role in it. Therefore, we have a clear clue where to put our OCG. At the same time the image classification problem is highly dependent on the holistic scene representation of the image. For example, a car is most likely seen in a street scene; similarly, a dining table is most likely seen in a kitchen scene. It has been already shown that the fixed-grid idea works fairly promising in classifying scene categorization problems [38]. Therefore, the image classification problem can best demonstrate the performance gain that we can get out of OCG method comparing to the fixed-grid alternative.

The OCG method can be used for scene classification problem as well. The drawback is that we cannot expect an object to be always present in some scene classes in general. Therefore, we may have no clear idea about where to put our OCG; unless we have a detector for some general objects like tree, sea, or road.

In this section we will first explain localization of OCGs according to the image frame and object bounding box. Following that, comes our feature representation. We will also specify technical settings of our BoF configuration such as size of our
visual vocabularies. Finally in 3.3.3, our classification approach is discussed.

### 3.3.1 Localizing The OCG

As we explained before, the idea behind OCG basically is to partition the image with respect to position and size of some object of attention. We want the center cell of the grid to be placed over the object so that the other cells model the relative distribution of surroundings of the object.

Therefore, in the images where the central object is present, we form the OCG in such a way that the center cell exactly fits bounding box of that object. During both training and testing, we use ground truth annotation of objects that are present in the image to form our OCGs. This is the only way we could measure the maximum amount of information gain we can expect from the OCG method.

If there are more than one object of the same class in the image, we extract an individual OCG feature based on each of the objects and classify them independently. Maximum of the classification outputs will then be set for classification confidence of the test image.

For negative samples, we do not have any objects of the corresponding class in the image and hence there is no bounding box annotation in the ground truth for that image. Now the question is how to form our OCG cells. The simplest solution is to randomly generate a set of bounding boxes from a uniform distribution and localize our OCG with respect to that.

According to [72], probability of having an object in a specific location of an image is not equally distributed over the image. The probability of observing an object in each pixel of an image can be estimated by the visual appearance of context of the image.

We train a probability distribution for the position of different objects in the image as well as the expected size of the object bounding box. Our intuitive approach, as we will shortly explain, is independent of image content. The trained distribution can, therefore, be used to generate random bounding boxes in negative images.

Our model for generating bounding boxes of objects is a multivariate gaussian distribution. The variables include x, y coordinates of the upper left corners of the bounding box as well as its length and height. We train this model by the bounding box annotations that we have for positive images in the training set.

Figure 3.3 illustrates the gaussian models that we trained for spatial distribution of bounding boxes of two classes of objects namely bottle and car.

The images of Figure 3.3 are obtained in the following way: Firstly all pixels of the image are set to zero. Then 100 bounding boxes are sampled from the gaussian model that has been trained for the spatial distribution of the object. Finally, for each sampled bounding box, the intensity value of all of the pixels inside the bounding box is increased by one. As you can see, the bounding boxes sampled from the spatial distribution of car are horizontally elongated. On the other hand, the sampled bounding boxes of bottle are elongated in vertical direction. 

32
3.3. OUR APPROACH

Figure 3.3: We model spatial distribution of different objects by multivariate gaussian distributions. (a) represents distribution of location, size and aspect ratio of 100 random bounding boxes sampled from spatial distribution of bottle. Similarly, (b) represents bounding boxes sampled from spatial distribution of car.

3.3.2 Feature Extraction

It is believed that contextual information, in general, can be represented by the following main types of features [72]:

**Distribution of structuring elements:** structure of the surrounding objects can help classification of context into different scene classes. *Gist* is a type of feature that is useful for representing contextual information of an image and it has been used for scene classification problems [72, 51, 71]. Gist, in fact, is the response of Gabor filters in several orientations and frequencies.

SIFT-like features have also been successfully used for modeling contextual information [38, 73]. In particular [5] has shown that Color-SIFT is the best performing feature type for classification of scene categories. Moreover, [18] and [5] have shown that dense features perform better than sparse sampling for scene classification problems.

In order to model the structural distribution of context we create a BoF histogram over a visual vocabulary of HOG features [13]. Local HOG features are similar to SIFT [42] in essence. The visual vocabulary is created by clustering HOG features using K-Means algorithm. In order to fill in the BoF histogram, several overlapping dense grids with multiple scales are defined on the image. For each grid cell, HOG feature is extracted. Then the cell is labeled according to the index of the nearest match in the visual vocabulary. Finally, we built the BoF histogram over the labels that were assigned to the dense grid cells. We refer to this HOG histogram as *shape-BoF*.

**Color distribution:** Distribution of color features can also be informative in scene classification. For example, the dominant color in a forest scene is expected to be green while a beach scene is more bluish.
We model the color distribution in two levels. In the first layer each pixel of the image is labeled based on its RGB values. We use K-Means to build the visual vocabulary of RGB values for the first layer. We call this vocabulary \( CV_1 \). During the first layer, we label all of the pixels of the image according to the \( CV_1 \).

Remember that, we defined a multi-scale dense grid on the image while building the shape-BoF. In the second layer of building color histogram of the image, we use the labels that we assigned to each pixel in the first layer to create a histogram for each cell of the dense grid. Obviously, length of this histogram is equal to size of \( CV_1 \). We use K-Means once more to create a new visual vocabulary using this histograms that we created per each cell. We call this new vocabulary \( CV_2 \). \( CV_2 \) is used to label the color histograms that we already build per cell.

The color distribution is, finally, modeled by building a histogram over the labels of the cells over the \( CV_2 \). We refer to this histogram as color-BoF.

**Semantic relations:** Co-occurrence of the set of objects that are present in an image can be considered as yet another source of information that can help us in scene classification. In [62], this type of information is used to re-rank the segment labels within an image. Moreover, one could also use spatial relations of the objects such as sky is on the top of sea, road is underneath car. Heitz et al. automatically train the optimal set of spatial relations between objects [32]. Nevertheless, [81] claims that mutual location of the objects as a high-level representation cannot do much more than the low-level features in recognition of context.

As mentioned before, we use fixed-grid idea as our baseline. Fixed-grid incorporates the spatial and co-occurrence relations in a very simple and naive way when the co-occurrence relations can be interpreted as follows: if sky is at the top row grid-cells, building at the middle row grid-cells, and road at the bottom row grid-cells, then the image most likely contains a street scene.

Our OCG method, exploits the semantic relationship between objects similar to the fixed-grid idea; we expect OCG to model the spatial information and semantic relations more coherently than the fixed-grid approach though.

### 3.3.3 Classification

So far in this section, we have talked about the type of features that we extract from a region of an image and the type of information that we get from those features. As showed in Figure 3.1, 3.2, images are partitioned into 9 cells; both in OCG and fixed-grid strategy. We calculate one shape-BoF and one color-BoF for each cell and then concatenate them altogether.

It has been shown that for the problem of scene classification SVM is the superior classifier compared to KNN [5]. Therefore we use SVM for classification of our feature vectors. We train a SVM with RBF kernel for each category of images using one-vs-all approach. Our experiments verify the fact that \( \chi^2 \) distance performs better than Euclidian distance in BoF framework [73, 34, 82]. In our scene classification problem, \( \chi^2 \) increased the average precision scores from 5 to 15 percent varying for different object classes comparing to Euclidian distance.
3.4 Experimental Results

As we mentioned before, we approach the image classification problem from scene classification point of view. The actual problem to be solved is to answer the question whether there is an instance of a specific object class (say car) in an image or not. The way we answer this question is to make decision based on the contextual information of the image with respect to specific objects.

3.4.1 Database

We experiment on PASCAL VOC 2007 database [17]. This is a general database and many different challenges are done on this database including image classification, object detection and object segmentation. This database has bounding box annotation for many types of objects which will help us to perfectly form our OCG features. The bounding box annotation belongs to object detection challenge though.

The database contains 20 different object classes in total. The images are divided into three disjoint sets namely train, validation, and test. the three data sets contain 2501, 2510, 4952 images respectively. We merge train and validation sets and use it for training.

Uijlings et al. [73], which we have compared our work with their results, have also experimented on this database.

3.4.2 Implementation Details

The dense grid which we use for feature extraction starts from $12 \times 12$ cells. Neighboring cells overlap by half the cell-size. We use 7 different scales with scaling factor of $\sqrt{2}$. For the baseline method (the fixed-grid strategy), first of all a 3 by 3 grid of equal cell sizes is defined on the image.

Size of our shape visual vocabulary is 1000. For the color features, size of $CV1$ (the first layer visual vocabulary) is equal to 128 and $CV2$ (the second layer vocabulary), is of size 1000. Therefore the appended feature vector that we get from each cell of the 3 by 3 fixed-grid (or OCG as well) is $1000 + 1000 = 2000$. It means that length of the final feature vector in fixed-grid representation is equal to 18000.

3.4.3 OCG Configuration

Our final dream is to extract OCG features considering a wide variety of possible objects in the center of the grid and then combine them altogether. That is how we could automatically exploit both the spatial information and the semantic relations of different objects in an image. But, we do not have ground truth annotation for generic types of objects (such as sky, road, building, or any other typical object that can be found in natural images) to form OCG based on them. Thus, for each object category of the image classification problem, we use bounding box of the target object, itself, to form our OCG. Therefore, the results that we report
here, show the maximum extent of information that can be captured by the OCG idea. In practice, one does not have ground truth annotation for test images. This information is not provided in the image classification challenge of the Pascal VOC. However, in the object detection challenge, all of the instances of the 20 object classes are annotated and the bounding box information for them is available.

In order to get a fair conclusion from comparison of fixed-grid and OCG, we exclude the center cell from the OCG cells. Therefore, we use the bounding box of objects only to put our OCG on the right position in the image. In other words, we only use context of the object to evaluate our OCG method.

The feature extraction mechanism for OCG-cells is the same as what we did for the fixed-grid; with the only difference that, here, the center cell is excluded.

The final feature vector of the OCG method is concatenation of the fixed-grid features and the features we get from the cells of OCG. Therefore, size of the OCG feature vector is equal to $18000 + 16000 = 34000$.

If in one image, there is no object of the target class available to form the OCG based on that, we generate a random bounding box within the boundary of the image and consider it as the annotation of the target object (section 3.3.1). In cases where we have more than one object bounding box in the image, OCG feature is calculated for each of the bounding boxes. Then they are individually classified by SVM. Maximum value of the SVM outputs is then taken as the confidence value for the image.

To train SVM parameters, we do grid-search on cost and gamma value of the libsvm toolbox [10]. Optimal parameters are found by cross-validation with 5 folds. The evaluations are done with Pascal VOC toolkit [17] codes.

Table 3.1 illustrates the evaluation results of our OCG framework compared to
3.4. EXPERIMENTAL RESULTS

Table 3.1: Average Precision values for fixed-grid and OCG compared to the results of Uijlings et al. [73]. The only difference between the Fixed-Grid column and Uijlings-All column is that the Fixed-Grid, unlike Uijlings-All, incorporates spatial information into its histograms. The OCG column can be seen as Uijlings-All combined with Uijlings-Context-Only in the sense that they both use the information of bounding box annotations of the objects from ground truth to exclude the object patch.

<table>
<thead>
<tr>
<th>Object Category</th>
<th>Uijlings All</th>
<th>Fixed-Grid</th>
<th>Uijlings Context-Only</th>
<th>OCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bicycle</td>
<td>46.2</td>
<td>50.4</td>
<td>17.8</td>
<td>66.45</td>
</tr>
<tr>
<td>Car</td>
<td>69.0</td>
<td>67.2</td>
<td>43.1</td>
<td>83.79</td>
</tr>
<tr>
<td>Cat</td>
<td>43.7</td>
<td>48.8</td>
<td>15.5</td>
<td>68.11</td>
</tr>
<tr>
<td>Chair</td>
<td>44.9</td>
<td>47.7</td>
<td>39.0</td>
<td>60.81</td>
</tr>
<tr>
<td>Horse</td>
<td>69.2</td>
<td>72.9</td>
<td>56.4</td>
<td>81.94</td>
</tr>
<tr>
<td>Motorbike</td>
<td>49.1</td>
<td>56.2</td>
<td>25.3</td>
<td>75.01</td>
</tr>
<tr>
<td>Person</td>
<td>79.2</td>
<td>80.4</td>
<td>61.6</td>
<td>89.89</td>
</tr>
<tr>
<td>Sheep</td>
<td>28.4</td>
<td>34.0</td>
<td>15.0</td>
<td>54.26</td>
</tr>
<tr>
<td>TV monitor</td>
<td>40.9</td>
<td>42.1</td>
<td>33.7</td>
<td>52.17</td>
</tr>
<tr>
<td>Mean AP</td>
<td>52.29</td>
<td>55.52</td>
<td>34.16</td>
<td>70.27</td>
</tr>
</tbody>
</table>

the fixed-grid method, as well as the results of [73]. The table shows that for all of the object classes, OCG remarkably outperforms the fixed-grid approach. According to the last row of the table, OCG performs almost 15% better than the fixed-grid approach in terms of average precision over all of the 9 object classes. However, as we mentioned before, this is the maximum extent of progress that one could expect to get using OCG instead of the fixed-grid method. That is because the OCG is ideally localized based on ground truth annotation which is not the case in a real case problem.

In the second column of Table 3.1, that is labeled as Uijlings-All, the image is represented by a BoF histogram of SIFT-like features. Uijlings-Context-Only is the same as Uijlings-All except from the fact that the object patch is removed from image before building the BoF histogram. It means that they use ground truth annotation to cut the object out of the image and therefore use only context of the object for classification.

Even though the goal of [73] has not been to beat the state-of-the-art methods, yet the effect of incorporating spatial information is obvious from the comparison of their results to our Fixed-Grid results.

Aside from some minor implementation details, their experimental setting is
very much similar to ours. Their evaluations is also done on Pascal 2007. They also use BoF framework as well as SVM classification. The only difference is that size of their visual vocabulary is 4096 while we use a vocabulary of size 1000, as mentioned before. The other difference is that they use SIFT-like features while we use HOG. However, we believe that using SIFT will further increase our performance [5].

Neglecting the aforementioned differences, there is a remarkable difference between the Uijlings-All column and the Fixed-Grid column in Table 3.1. Fixed-Grid is superior because of adding spatial information into the BoF histograms.

Table 3.1 shows that for all of the object classes, except car, the Fixed-Grid is exclusively outperforming the Uijlings-All results. The small loss of average precision for the Car class turns out to be because of the confusion between context of the car class with the bus class. Most of the high score false positives of the car class are bus samples (Figure 3.5). Uijlings et al. also mentioned in their paper that context of car is a superset of the context of bus [73].

We want to stress once more that even though we use the bounding box annotation of objects for our evaluations, but, the center cell is totally ignored in evaluations of OCG. Therefore, all the information that the OCG appends to the end of fixed-grid comes from the surroundings of the object which is considered as the holistic contextual information of that object.

To take the first step towards using a real detector instead of ground truth, we made yet another experiment simulating the behavior of a detector. Results of this experiment for the car and motorbike object class are shown in Figure 3.4. There are three curves for each object class in the figure. The difference between the curves is the number of random bounding boxes that we used when evaluating the method on a test image. In fact, these bounding boxes are used to form the OCGs. Therefore, for each of these random bounding boxes we get one OCG feature. We
showed the curves for 1, 10, and 20 random bounding boxes per image.

The random bounding boxes can be interpreted as false positives of the detector. The vertical axis shows average precision score. Along the horizontal axis we gradually add ground truth bounding boxes; thus the horizontal axis can be interpreted as the detection ratio of the detector.

The dashed line in Figure 3.4a says that if the detector produces 10 false positives per image (the red curve), and if the detector finds 30\% of total objects accurately, then the OCG method will start outperforming fixed-grid (cf. Table 3.1). The more number of object are detected correctly by the detector, the higher the performance of OCG will be compared to the fixed-grid baseline. The same situation happens for the motorbike class starting from 40\% recall of the detector (Figure 3.4b).

It is worth mentioning that for the class of car, there are 721 objects in the database in total. Thus, 10 false positives per image, which was the case in the aforementioned scenario, means that a detector with the precision value equal to $\frac{30\% \times 721}{4898 \times 10} \approx 0.004$ is sufficient; where 4898 is the number of clear test samples. Similarly, for the motorbike class a detector with precision $\frac{40\% \times 222}{4941 \times 10} \approx 0.001$ is sufficient.

It may seem reasonable that if we generate one random bounding box per test image, the classification accuracy should basically be almost equal to the fixed-grid result. But, comparing the result of Table 3.1 with the leftmost point of the green curve in Figure 3.4 shows that fixed-grid is strictly better than OCG in such a situation. We believe that this happens because most of the images in Pascal has the target object in the center of them. Therefore, it is almost always the case that the central object is in the center of the image and thus is better captured by the fixed-grid.

3.5 Conclusion and Future Works

In this chapter we proposed a new method for incorporating spatial information into BoF framework. The method, which we call it Object Centered Grid (OCG), works based on some central objects and will model the contextual information of an image based on the distribution of surrounding of those objects.

We showed that in the perfect case, when we have accurate localization of the central objects, the OCG framework captures contextual information in a much more coherent and stable form compared to the fixed-grid method.

The very next step of our work will be to integrate the OCG idea with a real detector and try to increase performance of any object detector with OCG. However, we believe that according to imperfectness of local object detectors, one have to be very careful in integrating a real detector into the OCG framework. Moreover, the false positives, that are subject to any typical object detector, could bring some problematic issues into question. However, in this chapter our main goal was to measure the amount of information gain that we can attain by using OCG in the ideal case.

Another direction to go in future is to study effect of fusing OCGs of different
objects together. It would also be interesting to make some experiments with OCG on scene classification databases. This later option requires some general object detector (such as sky, building, tree, road) or information about their bounding boxes to be available.

The ideas introduced in this chapter are published in [58].
Chapter 4
Attributed Local Features for Image Classification

Local descriptors have been shown to be very useful in many computer vision tasks including object recognition, image matching, geometric reconstruction, image retrieval, and action classification. Their expressive power is, however, limited in different ways. For example, when we put local features together in order to construct global representation of visual content we might end up merging incoherent local features. We propose to group local features extracted from image/video regions based on higher-level information. For example, region attributes such as textureness and surface orientation may help to disambiguate similar SIFT descriptors extracted from buildings and roads. Concretely, we define multiple sets of region attributes using geometric layout, unsupervised bottom-up segmentation, and fore/background decomposition. We use these attributes in order to disambiguately aggregate local features. Our approach can be seen as an extension of spatial pyramid matching by Lazebnik et al where image regions are labeled according to fixed spatial grids. More generally, it offers the possibility of benefitting from pre-learned knowledge within the paradigm of bags of features and their extensions. Despite all of these attributes being imperfect, we demonstrate that they provide complementary information to local features in a wide variety of tasks. We evaluate our method on PASCAL VOC 2007 image classification dataset, Oxford Buildings image retrieval dataset, and Hollywood2 human action dataset and show remarkable improvements in all of the tasks.

4.1 Introduction

In this chapter our goal is to improve discriminative power of local features by information extracted at a non-local image level. In particular we focus on attributes associated with image regions. Region properties have been widely explored in the past ranging from texture and color descriptors [3, 23] to object-level [67] and 3D surface attributes [33]. While some of the region descriptors can be defined in a
data-driven manner, other more sophisticated region properties can be learned from
the training data [23, 67, 33]. Image regions are usually defined by image areas
with regular properties [40]. Such properties are often observable at the region
level only and are not visible for local features. While image regions have been
traditionally designed as building blocks for higher level tasks such as object and
scene segmentation, here we propose to use region-level information to augment
local feature descriptors and to improve subsequent tasks of image classification
and retrieval.

We wish to use image regions which can be extracted consistently for image
structures with similar semantics in different images. As no perfect segmentation of
such kind is known, we resort to using multiple alternative segmentation strategies.
Among different possible segmentation methods we test (i) deterministic regions
defined by location in the image [38], (ii) data-driven bottom-up image segmentation
based on color and texture properties [21] as well as (iii) learned segmentation in
terms of Geometric Context [33] and object detection bounding boxes [20]. The last
alternative of using pre-learned segmentation methods is particularly interesting as
it enables us to introduce different types of knowledge learned on related tasks into
bag-of-features image representations.

Our method can be seen as an extension of Spatial Pyramid Matching method by
Lazebnik et al. [38]. The method in [38] attributes local image descriptors according
to their coarse location within the grid of spatially pre-defined image regions. This
simple approach has shown significant improvement for image classification tasks
due to the frequent correlation of locations and the meaning of objects in the image
(top/sky, bottom/road). Location-defined image attributes, however, often fail to
separate images into consistently meaningful parts. In this chapter we address
this problem and extend [38] by attributing local features according to the data-
dependent and semantically-consistent image segmentation.

Figure 4.1 illustrates the idea on an image matching example. We show matches
between two images using local features extracted from cars and trees in the first im-
age. Local descriptors alone result in many ambiguous matches (Figure 4.1(d)-(e)).
Separation of local features into three sets defined by spatial grids in Figure 4.1(b)
(similar to [38]) improves results for trees but generates many wrong matches for
cars (Figure 4.1(f)-(g)). Local features in Figure 4.1(h)-(i) are matched according
to local descriptors while respecting labels of Geometric Context image segmen-
tation [33] illustrated in Figure 4.1(c). As can be seen, this more semantically-
meaningful segmentation improves matching results significantly in this case. The
quality of feature correspondence is essential to BoF-based methods, hence, this ex-
ample motivates our approach aiming to combine information at the levels of local
features and image regions.

Our approach is related to the line of work on contextual object recognition
[71, 62]. While this work attempts to disambiguate object labels based on the
surrounding scene context, here, similar in spirit, we try to distinguish ambiguous
local descriptors based on the context of their regions.

The proposed method can be potentially interesting for many applications of
4.1. INTRODUCTION

Figure 4.1: Matching of local features between two different street scenes. The local features of cars and trees from the first image are matched to the densely sampled local features in the second image. The features are quantized according to the SIFT vocabulary with 4000 Visual Words (VW). Matches in (d)-(e) correspond to the pairs of features with the same VW labels. In (f)-(g) matching points share identical WV labels and region labels defined by the spatial grid in (b). In (h)-(i) the features are matched according to VW labels and region labels defined by Geometric Context [7] image segmentation illustrated in (c).
local features. In this chapter we validate our method on the tasks of image classification and image retrieval. In both cases we use Bag-of-Features (BoF) image representations and represent images by histograms of local SIFT features quantized according to the standard visual vocabulary and with respect to the labels of corresponding image regions. For image classification we use Support Vector Machines (SVM) trained and tested on the set of kernels corresponding to alternative image segmentations. For the task of image retrieval, we rank database images according to their distance to a query image in the BoF-space.

The rest of this chapter is organized as follows. In Section 2 we describe image representation in terms of quantized features as well as alternative image segmentation methods used in this chapter. Section 3 presents image classification and image retrieval. In Section 4 we present extensive experimental evaluation of the proposed idea while we conclude this chapter in Section 4.5;

4.2 Local Feature Encoding

In this work we aim to improve Bag-of-Features (BoF) image representations by attributing local features with region information. For image classification problems we build upon the standard local SIFT [10] features densely extracted from different locations and scales in the image\(^1\). We construct visual vocabulary of size \(N = 4000\) by KMeans clustering of 106 SIFT vectors extracted from training images. Each local feature is then KMeans quantized to the label \(w_i, i = 1..N\). For image retrieval problem we follow the setup of [5] and use SIFT descriptors extracted at affine-invariant Hessian regions [22] quantized with the dictionary of size \(N = 16\). Given segmentation of an image into a set of regions with \(M\) labels, we assign label \(r_j\) to local features inside region \(r_j, j = 1..M\). Finally, to measure statistics of features with labels \((w_i, r_j)\), we label features according to the full dictionary \(D\) with \(MN\) labels corresponding to all label pairs \((w_i, r_j)\). In the rest of this section we describe the four different methods for generating labeled image regions and the associated full dictionaries used in this work.

4.2.1 Spatial Grids

Spatial Pyramid Matching [2] introduced division of images into a set of predefined spatial grids. We follow this approach and define nine Spatial Grids (SG) as illustrated in Figure 4.2(top). Each of the nine SGs divides an image in up to \(M = 9\) regions with unique region labels. According to these SGs we define nine full dictionaries \(D_{SG_i\times j}, i = 1..3, j = 1..3\). Note that the first dictionary \(D_{SG_1\times 1}\) corresponds to Bag-of-Features (BoF) representation. We use BoF and SG representations as baselines for image classifications in this chapter.

\(^{1}\)We use the following parameters for the dense SIFT sampling: patch size \(d = sd_{min}\) with \(d_{min} = 12\) pixels; spatial step size \(\Delta d = d/2\); scale \(s = 2^i \times 4^i, i = 0..10\).
4.2. LOCAL FEATURE ENCODING

4.2.2 Quantized Regions

To generate data-driven image regions with specific region properties, we over-segment images into a set of initial regions (also known as superpixels) using the method and the code of [19]. To generate region descriptors, we then extract signatures of color and texture from each region. To represent color, we compute mean RGB values and hue histograms. For the texture, we compute mean absolute response and the histogram of maximum responses of a filter bank defined in [18].

Given two color and two texture descriptor vectors for each region, we construct visual dictionaries for these vectors independently using k-means with \( k = 2, 4, 8 \). As a result, we obtain 12 dictionaries which we use to label image regions and to generate 12 alternative image segmentations. Examples of such segmentation for regions defined by texture histograms and \( k = 4 \) are illustrated in Figure 4.2(middle). Based on the 12 alternative image segmentations we define full dictionaries denoted as \( D_{QR}, i = 1..12 \).

4.2.3 Geometric Context

Geometric context has been introduced in [7] as a means of segmenting images into regions and classifying them according to their coarse geometric properties such as 'support', 'vertical' and 'sky'. The segmentation and the labels of such regions have been learned on many labeled images in order to enable generalization of the method to new and diverse scenes. We use public implementation of geometric context provided by the authors and generate five alternative image segmentations with regions (i) support/no-support; (ii) vertical/non-vertical; (iii) sky/no-sky; (iv) support/vertical/sky and (v) support/left/center/right/porous/solid/sky (we refer to [7] for the details). Illustration of region segmentation according to geometric context is illustrated in Figure 4.2(middle). Based on the five types of segmentations we define full dictionaries \( D_{GC}, i = 1..5 \).

4.2.4 Object Detection

As an alternative type of image segmentation we also consider object bounding boxes provided by the object detector [14]. In this chapter we use detectors for 20 object classes trained on PASCAL VOC 2008 dataset. Each object detector defines a binary image segmentation mask with object/non-object regions. Segmentation results generated by the output of a car detector are illustrated in Figure 4.2(bottom). We use detector-based image segmentation to generate twenty full dictionaries \( D_{OD}, i = 1..20 \) corresponding to twenty object detectors.

\(^2\)Our two color and two texture descriptor vectors are computed using public implementation of Geometric Context [7].
Figure 4.2: Illustration of image segmentation according to (from top to bottom) Spatial Grids (SG), Quantized Regions (QR), Geometric Context (GC) and Object Detection regions (OD). For SG we show the nine types of fixed segmentations. For QR, GC and OD we show segmentation examples on three images. Illustrated QRs correspond to the texture regions quantized into four four labels. Illustrated OD regions have two possible labels (car/no car in this example).
4.3 Image Classification and Retrieval

Previous section has described alternative image segmentation methods. All segmentation methods define the total of 46 full dictionaries: $D_{SG_{i,j}}, i = 1..3, j = 1..3$, $D_{QR_{i}}, i = 1..12$, $D_{GC_{i}}, i = 1..5$, $D_{OD_{i}}, i = 1..20$. We use all full dictionaries for quantizing local image features 164 in all images. The resulting feature labels are used to generate 46 alternative histograms of quantized features for each image.

4.3.1 Classification

For image classification we use Support Vector Machines [23] (SVM). To combine evidence from alternative image descriptors above, we compute distances for each type of 46 histogram $l_1$-normalized vectors (we call them channels) independently. We use the multi-channel Gaussian kernel

$$K(x_i, x_j) = \exp\left(-\sum_c \frac{1}{\Omega_c} D(x_i c, x_j c)\right)$$ (4.1)

where $D(x_i c, x_j c)$ is the $\chi^2$ for the feature channel $c$, and $\Omega_c$ is the average channel distance [3].

To build a multi-class classifier we use one-against-rest strategy. To compare the overall system performance, we compute a mean Average Precision (mAP) over a set of binary classification problems. Different channel combinations will be described and evaluated in Section 4.1.

4.3.2 Retrieval

For image retrieval we follow the setup in [5] and compute the rank for each image in the database according to its distance from a query image. We use Euclidean distance to compare $l_2$-normalized histogram-based image representations, we also use tf-idf word weighting scheme. To benefit from alternative full dictionaries defined above, we sum image distances computed from alternative histogram image descriptors. For the retrieval problem addressed in Section 4.2 we only consider combinations of the original BoF dictionary with Geometric Context dictionaries $D_{GC_{i}}, i = 1..5$.

4.4 Experimental Evaluation

4.4.1 Image Classification

We evaluate our method on the image classification task of PASCAL VOC 2007 dataset [17] (refer to Section 3.4.1 for more information on the dataset).

In this section we present performance of different types of features previously introduced in Section 2. We also show the amount of improvement obtain by the combination of these features indicating complementary properties of local features quantized with respect to different types of image regions.
CHAPTER 4. ATTRIBUTED LOCAL FEATURES FOR IMAGE CLASSIFICATION

Figure 4.3: Classification performance of five different feature types are showed. 
*SG-1x1* is the basic approach in which histogram of local features over the whole 
image is used for classification. *SG-9channels* denotes to combination of 9 different 
types of spatial grids shown in Figure 4.2. *QR-12channels* refers to combination of 
features from 12 different types of quantized image regions. *GC-5channels* refers to 
combination of features based on geometric context segmentation and finally in *OD* 
we combine features from foreground/background regions segmented by a generic 
object detector.

Figure 4.3 illustrates relative performance of five different types of image features 
obtained from different image segmentation strategies. The performance of baseline 
BoF method (denoted as SG-1x1) is inferior compared to the other four methods 
using alternative image regions for feature quantization.

Figure 4.4 illustrates comparative per-class improvement of average precision 
with respect to the spatial grids method. We can observe that using different 
region segmentation results in improvement of different object classes. For example, 
bicycles are classified better when using OD segmentation while for plants Quantized 
Region segmentation is more helpful. For tables, on the other hand, Spatial Grids 
perform the best. Therefore, we next combine different sets of image features.

Figure 4.5 illustrates performance of feature combination corresponding to the 
combination of two or more image segmentation strategies. It is clear from the 
Figure that the combination of all different types of image features (denoted by 
*SGs+QRs+GCs+od* in the figure) results in a consistent improvement of any of 
individual features for most of the object classes. Table 4.1 compares different image 
classification methods based on their mean average precision. Methods reported in
Figure 4.4: Improvement obtained per-class with different image segmentation strategies compared to SQ-9channels. Object classes in each plot are sorted based on the decreasing amount of improvement.
CHAPTER 4. ATTRIBUTED LOCAL FEATURES FOR IMAGE CLASSIFICATION

Figure 4.5: Classification performance for the combination of alternative image segmentation strategies. $SGs+QRs$ and $SGs+GCs$ represent combination of the 9 spatial grids with Quantized Regions (12 channels) and Geometric Context (5 channels), respectively. $SGs+od$ represents combination of features from segmentation provided the object detector and by the spatial grids. $SGs+QRs+GCs+od$ illustrates combination features corresponding to all different types of segmentation strategies. We show baseline performance of BoF (SG-1x1) and spatial grids (SG-9channels) for comparison.

the first part of the table use no external annotation apart from 215 the image labels for the training set. In the second part, although the image classifier is still trained on PASCAL 2007 images and annotations, the images are segmented using some tools which have been trained on additional training data. The performance of the two best methods on PASCAL 2007 challenge is also reported in the last part of Table 4.1 for comparison. From the table we observe significant improvement provided by both supervised and unsupervised image segmentation.

Our method uses only one type of local features compared to the best performing method based on multiple local features. While our intention in this work was to demonstrate advantage of region attributes for BoF representations, the performance of our method is expected to improve further if combined with multiple local detectors. Our main contribution in this chapter is that we show consistent and considerable improvement over a reasonably powerful and general baseline using a novel method.
4.4. EXPERIMENTAL EVALUATION

Table 4.1: Overall mAP performance by different methods used in this chapter as well as by the best performing methods (INRIA) in PASCAL VOC 2007 image classification challenge.

<table>
<thead>
<tr>
<th>Image Features</th>
<th>Performance (mean AP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detector-only</td>
<td>42.30%</td>
</tr>
<tr>
<td>BoF</td>
<td>53.30%</td>
</tr>
<tr>
<td>Spatial Grids (1x1 to 3x3)</td>
<td>56.52%</td>
</tr>
<tr>
<td>Quantized Regions (12 channels)</td>
<td>54.67%</td>
</tr>
<tr>
<td>Spatial Grid + Quantized Regions</td>
<td>57.45%</td>
</tr>
<tr>
<td>Geometric Context (5 channels)</td>
<td>55.42%</td>
</tr>
<tr>
<td>Object Detector segmentation</td>
<td>56.47%</td>
</tr>
<tr>
<td>Spatial Grid + Geometric Context</td>
<td>57.12%</td>
</tr>
<tr>
<td>Spatial Grid + Object Detector</td>
<td>58.50%</td>
</tr>
<tr>
<td>Spatial Grid + Geometric Context + Object Detector</td>
<td>59.90%</td>
</tr>
<tr>
<td>Quantized Regions + Object Detector except from target object</td>
<td>57.06%</td>
</tr>
<tr>
<td>INRIA FLAT (2nd best in VOC’07)</td>
<td>57.46%</td>
</tr>
<tr>
<td>INRIA Genetic (VOC’07 winner)</td>
<td>59.39%</td>
</tr>
</tbody>
</table>

4.4.2 Image Retrieval

We evaluate performance of image retrieval on Oxford Building dataset containing 5062 images and 55 queries [60]. We use quantized features provided by the authors of [60] and report performance in terms of mAP, i.e. mean average precision over 55 query runs. Using the standard BoF representation we obtain mAP=61.5 which matches closely to the corresponding mAP=61.8 reported in [60]. Using our methods with attributed local features and Geometric Context dictionaries $D_{GC_i}$, $i = 1..5$ (cf Section 3.2) we are able to improve BoF baseline to mAP=62.5. Figure 4.6 illustrates an example of a false match which was ranked high by the BoF method. Our method was able to correct the rank of this match from 22 to 52 by disambiguating incorrect matches between the building and the tree.

Although our improvement is not very high in comparison to mAP=0.647 reported for the spatial verification step in [60], we note that spatial verification is a costly operation which cannot be performed efficiently on all image pairs. On the contrary, our method is of approximately the same cost as the baseline BoF approach and can be used to improve the initial ranking of images.
CHAPTER 4. ATTRIBUTED LOCAL FEATURES FOR IMAGE CLASSIFICATION

Figure 4.6: Example of a false match from the Oxford Building dataset. The original rank of this match (22) obtained with the baseline BoF method was improved by our method by lowering the rank of this false positive to 52.

4.5 Conclusion and Future Works

We have introduced a notion of attributed local descriptors by adding information from image regions to local features. We have investigated alternative types of image regions defined both bottom-up and learned on related tasks. In both these cases our method has demonstrated a significantly improved performance on the image classification task while using only a single type of local feature descriptor. We have also demonstrated improvement by our method for the image retrieval task. The proposed approach can be easily applied to other applications of BoF representations. Moreover, it provides a conceptually simple and efficient way of integrating prior knowledge (possibly learned on related tasks) with BoF-based methods.
Part III

Disambiguating Local Features in Videos
Chapter 5

Spatiotemporal Aware Representations for Human Action Recognition

In this chapter we study importance of visual contextual information in recognition of actions both in terms of motion features as well as appearance features. We follow the OCG idea introduced in Chapter 3 and extend it to videos. We make an empirical evaluation of the state-of-the-art human action recognition methods and analyze effects of spatial and temporal alignment of actions on the recognition performance. We also provide full bounding box annotation for one of the most recent human action recognition databases.

5.1 Introduction

Due to rapid growth in popularity of video sharing websites problem of Human Action Recognition in realistic scenarios has become a hot topic in Computer Vision. The main problem is to, based on a sequence of image frames, i.e. a video, answer the question of which action class is performed within the video. The big challenge is that realistic human actions have a wide variety in terms of motion, background/foreground appearance, viewpoint, etc. Results reported in [16, 52, 65] show impressive recognition performance of human actions in videos. More interestingly, the solutions are good enough to yield promising performance on real movies as well [37, 36, 47].

Some of the action classes may be context-free which means that inclusion of the scene in which the action is performed brings no extra information about the action class as in HugPerson action. This action could be done within very many different scene contexts and therefore inclusion of context is redundant. In general, it is expected to observe a strong correlation between type of the scene in which the action is performed and the action itself. For example sitUp action most likely happens in a bedroom scene context [47] and GetOutCar action is characterized by a car in close proximity of the actor [30]. Therefore, analogous to what we observed in Chapter 3, we use contextual information to be useful in boosting up accuracy of
Figure 5.1: At the left side a sample video clip is showed. We compare performance of five different action recognition approaches. In Approach{I-IV} the clip is classified based on exact temporal extent of the action while Approach-V extract features from the whole clip.

action recognition methods [41, 47]. In this chapter, we incorporate scene features with motion features and study mutual effect of them. We compare performance of five different approaches shown in Figure 5.1. A sample video clip is showed in the figure and the sequence of frames where action of interest is taking place is distinguished in yellow. We will explain more about configuration of any of these five approaches in Section 5.4.

Our main contributions in this chapter are threefold. First, we make an empirical study on several different types of features and analyze their relative importance. Second, unlike the conclusion made in [47], we show that visual context can help recognition of realistic human actions if modeled properly. Third, we provide public access to full annotation of actor bounding boxes in HOHA2 action database [47] which is the most recent and complete databases available for realistic scenarios.

The rest of the chapter is organized as follows. In Section 5.2 we study related works. Implementation details of classification method as well as type of feature descriptors that we use are illustrated in Section 5.3. Experimental results and configuration of our evaluations are specified in Section 5.4. In particular, Section 5.4.2 goes through annotation of actor bounding boxes in database. This data will be available for public use after the chapter is published. We conclude this chapter in Section 5.5.

5.2 Related Work

It has been showed that contextual perceptions convey a rich constellation of information. For example, using context, one can roughly estimate location and size of objects [72]. Context can also help image classification [?] and object detection accuracy [14]. Nevertheless, the problem of contextual priming is less studied in action recognition frameworks. Marszalek et al. [47] classify natural scenes into 10
different classes and show that even without having any representation of motion features at all, visual context of video frames can classify realistic human actions much better than a random classifier. Their experiments on action recognition from realistic movies showed that textual information retrieved from movie scripts gives a better description of scenes compared to visual representations. While this textual information is noisy, equivocal and subject to temporal misalignment incurred by lack of accurate temporal synchronization in movie scripts. On the other hand previous researches on scene classification in still images have demonstrated impressive performance on tens of different scene categories [18, 38]; yet without the temporal information which is also available in video. Accordingly, as we will demonstrate in this chapter, there should be a better way of capturing and representing visual information of context in videos which could result in integral improvement of the current action recognition methods. However, the primary goal of [47] is not to integrate contextual information into the problem of human action recognition but to provide a fully automatic framework for recognition of human actions in real movies. Their method is closest to our Approach-V in Figure 5.1.

In [30] the authors showed that using off-the-shelf person detectors one could gain much more accuracy in recognition of realistic actions compared to the former state-of-the-art method [36]. In fact they represent contextual information in terms of relative normalized location of detected persons as well as coexistence of people in proximity of each other as in HandShake and HugPerson actions. This approach also leaves a great portion of visual contextual information, being the holistic scene, untapped. Niebles et al. [52] proposed a probabilistic approach which uses both action and context information. But, they have not evaluated their method in realistic scenarios.

In some specific cases such as sports actions, integrity of contextual information is much more obvious which makes the problem less challenging though. In [41], generative graphical models are used to classify sports events and label objects and scene regions in still images. In their work, there is no motion feature involved and variety of scenes is restricted to some extent. Intuitively, for the case of realistic human actions there should be strong correlation between the actions and the scene in which the action is taking place. However, variety of motion and appearance in realistic scenarios is such wide that makes the problem very challenging.

Recall from Chapter 3 that we used BoF histograms to capture contextual information in images. There we showed that scene of images can be modeled much more discriminatively if we organize the histograms relative to some central objects such as car and people. This will help to compensate for the lack of spatial information which is the main drawback of the methods built upon histograms of local descriptors, in general. As we will see later, we have bounding box of the actors annotated for our database, therefore one could easily use the same idea by considering the central object to be the actor bounding box. In fact the method used in [?] can be considered as an extension of our Approach-III in Figure 5.1.
CHAPTER 5. SPATIOTEMPORAL AWARE REPRESENTATIONS FOR HUMAN ACTION RECOGNITION

5.3 Feature Types and Recognition Approach

Most of the previous works on realistic Human Action Recognition are based on BoF framework as in [37, 36, 47, 52, 30]. Therefore, we also build upon BoF idea to approach our problem.

5.3.1 Feature Descriptors

We use three types of features; SIFT [43], HOG [13], HOF [36], and HOGHOF which is concatenation of HOG and HOF features.

**SIFT:** We use ColorDescriptor software package [74] to extract dense SIFT features in 8 different scales with scale factor $\sqrt{2}$. The features are then quantized based on a vocabulary of features of size 4000. First we expand each video file into its constituent image frames and then extract SIFT features for each 2D image independently.

**HOG and HOF:** We use 3x3x2 spatiotemporal grids instead of 2D grids to compute HOG features. 3D HOG features, unlike SIFT, capture shape features which are dynamic in their local temporal neighborhood [47]. HOF (Histogram of Optical Flow) features are also extracted from local video volumes. HOF basically captures local motion information. We extract HOG and HOF from spatiotemporal interest points. To detect the interest points and to extract the features we use STIP software package\(^1\) which is a publicly available implementation of [35]. The code does not support automatic scale selection. It detects interest points at multiple spatial and temporal scales though. It is shown that in practice multi-scale features generally work better for many computer vision applications including Human Action Recognition in realistic scenarios [36]. After the features are extracted we quantize them based on a vocabulary of features of size 1000.

**HOGHOF:** Length of the HOG and HOF feature histogram is 72 and 90 respectively. HOGHOF is simply concatenation of the two and is a 162 bin long vector. This vector is then quantized based on a vocabulary of features of size 4000.

5.3.2 Classification Approach

Support Vector Machines have shown strong success in many Computer Vision applications. The state-of-the-art Human Action Recognition method [47] also benefits from classification power of SVMs.

$$f(X) = \sum \alpha_i y_i K(X^i, X) + b$$  \hspace{1cm} (5.1)

Equation 5.1 shows how SVM classifies features extracted from a given sample (denoted by $X$). In this equation, $\alpha_i$s and $b$ are constant values learnt during training stage of SVM, $X^i$ is the features extracted from the $i^{th}$ training sample and $y_i$ represents class label for the $i^{th}$ training sample. $K$ represents the kernel function used in the SVM classifier.

\(^1\)http://www.irisa.fr/vista/Equipe/People/Laptev/download.html
5.4 Experimental Results

5.4.1 Database

We use the challenging Hollywood 2 Human Action database (HOHA2) introduced in [47]. There are 12 classes of actions in the database. Some sample frames of the dataset are shown in Figure 5.2. The database contains 823 sample clips for training and 884 samples for test which are taken from 69 Hollywood movies. Training and test samples are taken from different movies.

There is an alternative training set in the database too which is automatically extracted from the movies using video scripts and subtitles and therefore is noisy. However, we only use the clean training set in our evaluations.

5.4.2 Annotation of Actor Bounding Box

As we already showed in Figure 5.1, for Approach-{II-IV} we need to know spatial location of actor during the whole temporal domain of actions. Therefore, we annotate the actor bounding box in all of the training/test samples in the database. Overall, we have 1322 annotations in the training set and 1440 annotations in the test set. Each annotation represents a new sample clip with accurate temporal and spatial extent. For clips which have more than one annotation associated with them, maximum response of action classifier is returned as classification score. Temporal extent of each annotation corresponds to the period of time that the action is taking place and its spatial extent corresponds to bounding box of the actor body. Actor bounding box is hand-labeled for some key-frames in videos and for the rest...
### Table 5.1: Action classification performance of different approaches on HOHA2 database.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean AP</td>
<td>57.82</td>
<td>53.38</td>
<td>55.31</td>
<td>43.88</td>
</tr>
<tr>
<td>AnswerPhone</td>
<td>30.82</td>
<td>27.61</td>
<td>24.40</td>
<td>21.62</td>
</tr>
<tr>
<td>DriveCar</td>
<td>89.92</td>
<td>88.71</td>
<td>91.17</td>
<td>73.84</td>
</tr>
<tr>
<td>Eat</td>
<td>40.44</td>
<td>39.38</td>
<td>42.83</td>
<td>33.48</td>
</tr>
<tr>
<td>FightPerson</td>
<td>76.57</td>
<td>73.01</td>
<td>73.90</td>
<td>66.36</td>
</tr>
<tr>
<td>GetOutCar</td>
<td>51.16</td>
<td>43.32</td>
<td>42.97</td>
<td>43.35</td>
</tr>
<tr>
<td>HandShake</td>
<td>46.77</td>
<td>31.98</td>
<td>44.37</td>
<td>29.20</td>
</tr>
<tr>
<td>HugPerson</td>
<td>46.64</td>
<td>47.21</td>
<td>43.63</td>
<td>18.08</td>
</tr>
<tr>
<td>Kiss</td>
<td>60.59</td>
<td>57.69</td>
<td>57.38</td>
<td>51.95</td>
</tr>
<tr>
<td>Run</td>
<td>79.61</td>
<td>73.31</td>
<td>82.17</td>
<td>58.61</td>
</tr>
<tr>
<td>SitDown</td>
<td>73.35</td>
<td>69.26</td>
<td>71.13</td>
<td>56.53</td>
</tr>
<tr>
<td>SitUp</td>
<td>25.55</td>
<td>23.10</td>
<td>20.77</td>
<td>20.28</td>
</tr>
<tr>
<td>StandUp</td>
<td>72.36</td>
<td>65.97</td>
<td>68.96</td>
<td>53.24</td>
</tr>
</tbody>
</table>

of the frames we estimate it from previous and next key-frame by a simple linear extrapolation.

### 5.4.3 Evaluations

In Table 5.1 we present performance of Approach-{I-IV} (see Figure 5.1). For 8 action classes (out of 12), Approach-III (Actor+Context) outperforms the other three approaches. This consistent improvement shows that visual context strongly supports recognition of human actions in realistic scenarios. It is interesting to note that the only actions for which Context features outperforms Actor features are DriveCar and SitUp. This supports the conclusion made in [47] regarding strong correlation between these two actions and their scene context. It is also interesting to see that for both of these two action classes (and also some others) combination of the two channels, i.e. Actor+Context, is remarkably better than either of them alone. It implies the fact that actor bounding box and context contain complementary information.

In Figure 5.3 we show that temporal alignment (represented by Time Aligned) does not improve classification accuracy compared to the case where temporal extent of actions is not well aligned (represented by Whole Clip). On the other hand, inclusion of contextual information, (represented by Actor+Context), improves the results considerably. We believe that the sudden deficiency of Actor+Context method in Eat action is because of the fact that in this case usually there are several persons doing the same action (e.g. eating on a lunch table). Therefore, when the action is not temporally aligned (in Whole Clip), other persons also support the action while
in our Actor+Context approach action of each actor is classified individually. Figure 5.4 illustrates relative contribution of 8 different information channels (4 coming from actor bounding box and 4 coming from context) in Approach-III. It is interesting to note that for Run action, motion information of context (HOF-context channel) is futile. While, on the other hand, motion of actor bounding box is the most crucial channel. This is also supported by results of Table 5.1 where Approach-II outperforms Approach-IV. Mean column in Figure 5.4 shows average contribution of different information channels over all of the action classes. It demonstrates that motion features of actor bounding box (represented by HOF-actor) are of the most crucial importance while motion features of context has the least contribution.

5.5 Conclusion and Future Works

We showed that classification of human actions in realistic scenarios is affected much more by spatial extent of the actor (actor bounding box) rather than temporal extent of the action. We also showed importance of contextual information in classification of human actions. However, goal of this chapter was not to propose an optimal solution for representing contextual information but to demonstrate that visual context produces an integral improvement in recognition of realistic human actions in a general framework. We showed relative contribution of different features coming from actor bounding box and context in performance of the combined classifier. We believe that extending the context modeling approach of [?] upon our actor bounding boxes will improve the results yet more.

Figure 5.3: Effect of spatial and temporal alignment of actions in classification performance.
CHAPTER 5. SPATIOTEMPORAL AWARE REPRESENTATIONS FOR HUMAN ACTION RECOGNITION

Figure 5.4: Relative contribution of different information channels in the final combined classifier. The values correspond to $\eta_m$s in Equation ???. Note that Mean column shows that HOF-actor and HOF-context are respectively the most and the least effective channel. This image is best seen in color.
Chapter 6

Attributed Local Features for Human Action Recognition

Local space-time features have recently shown success for action recognition within the Bag-of-Features (BoF) framework. But, as we discussed earlier, pure local descriptors have limited discriminative power resulting in the ambiguity of local features at different events. In this chapter we propose to disambiguate local space-time features and to improve action recognition by integrating additional supervision and non-local cues into BoF representation. For this purpose, we decompose video into different types of regions with specific properties. In particular, we investigate supervised and unsupervised video segmentation according to the estimated locations of motion regions, people, specific objects and actions. Despite this segmentation being imperfect, it provides complementary information for local features. We demonstrate how this information can be integrated with BoF representations within a kernel combination framework. We evaluate our method on the recent and challenging Hollywood2 action dataset and demonstrate significant improvements.

6.1 Introduction

Local video descriptors in combination with Bag-of-Features video classification have been shown successful for the task of action recognition [15, 36, 47, 52, 65, 78]. Local descriptors, however, need to be discriminative enough to overcome irrelevant variations in the video due to e.g. camera motion, lighting changes, projective effects and background clutter. As stated before, limited discriminative power of local descriptors may imply ambiguity of video representations and the resulting decrease of recognition performance.

The main goal in this chapter is to improve discriminative power of local video features by integrating non-local cues available at the region-level of the video. For this purpose, we propose to decompose video into regions with specific non-local properties and to disambiguate local features based on region labels. We wish to use regions which can be extracted consistently for video structures with
similar semantics in different videos. As no perfect segmentation of such kind is known, we resort to using multiple alternative segmentation strategies. A simple example of video decomposition we investigate in this chapter is the bottom-up motion segmentation. Motion-based foreground-background separation resolved at the non-local level can potentially add information to local features. Other examples of video decomposition include separation of video into foreground corresponding to people, faces or particular objects which can be obtained using object detectors pre-trained from still images.

Using different types of regions we construct alternative video representations from the original set of local spatio-temporal features. We exploit complementarity of these representations and combine them within a multi-channel SVM framework. We evaluate our method on the challenging Hollywood2 human action dataset and demonstrate significant improvement with respect to the state of the art.

6.2 The BoF framework

We follow previous methods for action recognition [36, 65] and use Bag-of-Features (BoF) video representation with multi-channel kernel SVM [82] for classification. In the following we briefly describe our framework and introduce ERC-Forest [50] for supervised learning of visual dictionaries for action classification.

6.2.1 Features

To extract local features in video we use on-line implementation of Spatio-temporal Interest Points (STIP) [35] combined with HOG/HOF descriptors [36]. Local features are extracted at multiple scale levels in space-time video pyramid. Each local feature is associated with HOG and HOF descriptors computed in the local space-time neighborhood of each feature and corresponding to position-dependent histograms of spatial gradient and optical flow respectively. We concatenate HOG and HOF descriptors into a single vector describing appearance and motion of local neighborhoods.

6.2.2 Histogram Representation

In the BoF framework, a video sequence is represented as a normalized histogram of local space-time features. The histogram is computed over labels (or visual words) associated with each local feature. Feature labels are commonly obtained by quantizing local feature descriptors according to a pre-learned dictionary. Following previous work [12, 36, 47, 69, 78], we first construct a visual dictionary using K-Means with \( K = 4000 \) visual words. While K-Means is a simple and unsupervised approach to construct visual dictionaries, previous work [26, 50] aimed to improve image classification tasks by constructing supervised dictionaries. Here we follow [50] and use ERC-Forest to construct supervised visual dictionary for action classification.
6.3. IMPROVING BOF WITH NON-LOCAL CUES

ERC-Forest is an ensemble of randomly created clustering trees [50]. It predicts class labels $c$ from local feature descriptors $d$. It benefits from labeled training set $L = \{(d_n, c_n), n = 1...N\}$ with $N$ descriptors $d$ associated with class labels $c$ and recursively builds random trees in a top-down manner. At each node, the labeled training set is divided into two halves such that the classes are separated well with the Shannon entropy maximized:

$$S_c(L, T) = \frac{2I_{C,T}(L)}{H_C(L) + H_T(L)}$$

(6.1)

where $H_C$ denotes the entropy of the class distribution in $L$, $H_T$ is the split entropy of the test $T$ which splits the data into two partitions, and $I_{C,T}$ is the mutual information of the split (see [50] for further details). Following [50] we use ERC-Forest to build supervised visual vocabulary for local space-time features. We construct $M = 5$ multiple trees with 1000 leaf nodes each and assign $M$ labels to each local feature according to each tree. ERC-Forests have been previously employed for image classification [39, 50, 53]. In Section 4 we demonstrate ERC-Forest to improve action recognition performance compared to K-means visual vocabulary. In Section 3 we introduce further improvements to visual vocabularies using non-local information associated with video regions. For each new vocabulary we construct a separate histogram representation of the video denoted by channels in this chapter.

6.2.3 Mutli-channel SVM

For action classification we use non-linear Support Vector Machine (SVM)[10] with RBF kernel. To investigate combination of different channels, we use multi-channel kernel [82] 4.1 where $x_i^c, x_j^c$ are histogram representations of videos $i, j$ using feature channel $c$. For multi-class classification we use one-against-all approach.

6.3 Improving BoF with Non-local Cues

This Section presents our extension to the BoF approach for action classification. We first present motivation for our method and then explain its details.

6.3.1 Motivation

Local descriptors should balance the trade-off between discriminative power and invariance required to overcome irrelevant variations of visual data. This poses a limit on the discriminative power of local features and implies their ambiguity. For instance, Figure 6.1 (top left) illustrates local video features with similar appearance but different semantic meaning (i.e. features extracted from the face and the door). Such ambiguous features aggregated within the BoF approach are likely to be assigned to the same visual word. BoF recognition implicitly relies of the correspondence of local features and potentially suffers from ambiguous features.
We propose to improve BoF action recognition and to disambiguate local features using higher-level information extracted from semantic video regions. By semantic regions we denote spatio-temporal regions in video with specific properties which are not available at the local feature level. For example, regions R1-person and R2-background (Figure 6.1, bottom left), when combined with local features (Figure 6.1, top left), help to differentiate the features detected on the person from those detected on the background (disambiguated features are illustrated with yellow and red circles in Figure 6.1, center). The BoF (frequency histogram) representation for such semantically disambiguated features carries additional information and therefore has the potential to improve upon the conventional approach. In this chapter we investigate different types of semantic video decomposition and their effect on action recognition.

### 6.3.2 Enriching BoF Representation with Region-level Information

In this section we describe alternative methods for decomposing video into regions and disambiguating local features. Given video segmentation into a set of regions with M labels, we assign label $r_j$ to quantized (with a visual dictionary D) local descriptors that fall inside region $r_j$, $j = 1, ..., M$. We encode this region-level segmentation of local descriptors within BoF representation by constructing a separate feature histograms for each region $r_j$ and concatenating such histograms into a single descriptor vector (or channel), cf. Figure 6.1, right. In the rest of this section, we describe different alternative strategies to extract semantic regions from a video.
6.3. IMPROVING BOF WITH NON-LOCAL CUES

6.3.3 Spatio-temporal Grids

Spatio-temporal video grids were introduced in [36] and showed promising results for action recognition. The basic idea is to divide a video into a set of predefined spatio-temporal regions. We follow the same approach and define 24 different spatio-temporal grids. Each of these 24 grids divides a video in up to $M = 27$ regions with unique region labels. The feature histograms corresponding to each spatio-temporal grid regions are then concatenated into one vector and normalized to make a channel. Spatially, we use a $1 \times 1$ grid (corresponding to the standard BoF representation), a $2 \times 2$ grid, a horizontal $h3 \times 1$ grid, a vertical $v1 \times 3$ grid, a denser $3 \times 3$ grid and a center-focused $o2 \times 2$ grid where neighboring cells overlap by 50% of their width and height. Temporally, a video sequence is divided into 1 to 3 non-overlapping temporal bins, resulting in $t1$, $t2$ and $t3$ binnings, where $t1$ represents the standard BoF approach. There is also a center-focused $ot2$ grid. In the following, we will refer to the combination of these 24 spatio-temporal grids as $STG24$ channel. Figure 6.2 illustrates some of the grids which showed good performance in [36].

6.3.4 Foreground/background Motion Segmentation

Segmenting local descriptors based on the foreground (FG) and background (BG) motions in a video can be valuable in order to separate foreground features which are more likely to belong to the action from background features which can help action recognition by capturing scene context. We use the Motion2D library [54] to estimate 2D parametric motion model in a video sequence. We then threshold (with four threshold values: 127, 150, 170, 200) the motion estimations and generate FG/BG masks. We use these masks to segment local descriptors into foreground and background classes. Figure 6.3 (1st column) shows the FG masks in green together with the segmented features. By separating features and building feature histograms according to foreground and background regions as well as for four different threshold values, we obtain 8 channels. We will refer to the combination of these eight channels as MS8 channel.
Figure 6.3: Illustration of proposed semantic region extraction in video according to (from left to right): motion region segmentation, action detection, person detection and object detection. Correct segmentation separates local features into meaningful groups denoted by yellow and red crosses. We also illustrated failures of automatic segmentation due to false negative detections (see e.g. missed running action in the first row) and false positive detections (see e.g. incorrect table detection in the third row).

### 6.3.5 Action detection

The ability to localize action in a video can be helpful in separating action specific descriptors and thus building a more representative BoF representation for that particular action. Of course, all the remaining descriptors that belong to the background of action, can form another complementary channel by capturing the context information.

The idea is to train an action specific detector on still images collected from the Internet and perform action detections on the Hollywood2 video sequences. Depending upon the availability of sufficient amount of action samples on the Internet, we investigate the idea for the following action classes: answering the phone, hugging, hand shaking, kissing, running, eating, driving a car, and sitting on sofa/chair. The last class corresponds to the action classes: sitting down, standing up, and sitting up. Figure 6.4 presents sample images collected from the Internet. We train Felzenszwalb’s object detector [19] for each action class (training on 100-170 true positives, and about 9000 true negatives) and perform separate detections on the Hollywood2 sequences (see Figure 6.3, 2nd column). We then threshold (with six threshold values: -1:25, ..., 0) bounding box detections based on the detector’s con-
6.3. IMPROVING BOF WITH NON-LOCAL CUES

(a) Object Detection

Figure 6.4: Sample images collected from the Internet used to train the action detectors.

...for each bounding box detection of an action at frame \(t\), we separate all the local descriptors that fall into that bounding box and are in the temporal range of \(t3\). We further divide each bounding box into a \(1 \times 1\) and \(2 \times 2\) grid (no grid for the background). For the \(1 \times 1\) grid, we simply generate the standard BoF representation (action specific channel). Whereas, for the \(2 \times 2\) grid, we compute a separate BoF representation for each of the \(2 \times 2\) grid cells and then concatenate all the 4 BoF representations into one vector to form another action specific channel. Consequently, for 6 threshold values and two grids, we end up having 12 FG (action specific) channels and 12 BG channels. We then concatenate the corresponding FG and BG channels into one vector and get 12 channels in the end. We will refer to their combination as AD-class channel for each of the nine aforementioned action classes, and as AD9 channel for the combination of all the nine AD-class channels.

6.3.6 Person Detection

Separation of local descriptors on the basis of person/non-person region segmentation not only helps to disambiguate them but also to provide a compact BoF representation for an action (as actions are related to persons). We use the Calvin upper-body detector [1] which is a combination of the Felzenszwalb’s object detector [19] and the Viola-Jones’ face detector [77]. This detector returns bounding boxes fitting the head and upper half of the torso of the person (see Figure 6.3, 3rd column). Following the steps of Section 6.3.5, we come up with 12 channels. We will refer to their combination as UB channel.

\(^1\)Based on a cross validation experiment, it turns out that the concatenation of FG and BG channels performs better than treating them as separate channels.
CHAPTER 6. ATTRIBUTED LOCAL FEATURES FOR HUMAN ACTION RECOGNITION

Table 6.1: Overall performance by the baseline channels.

<table>
<thead>
<tr>
<th>Channels</th>
<th>Performance (mean AP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoF with K-Means</td>
<td>47.91%</td>
</tr>
<tr>
<td>BoF with ERC-Forest</td>
<td>48.55%</td>
</tr>
<tr>
<td>STG24 with K-Means</td>
<td>50.42%</td>
</tr>
<tr>
<td>STG24 with ERC-Forest</td>
<td>51.83%</td>
</tr>
</tbody>
</table>

6.3.7 Object Detection

Objects can provide a valuable context information in recognizing actions in video. For instance, the object car can be helpful to recognize the actions driving a car and getting out of a car, and the objects chair and sofa can be helpful for the classes sitting down and standing up. We investigate this concept by using Felzenszwalb’s object detectors [19]** on the following object classes: car, chair, table and sofa and perform separate detections on the Hollywood2 sequences (see Figure 6.3, 4th column). Again following the steps of Section [?], we compute 12 channels. We will refer to the combination of these 12 channels as OD-car/chair/table/sofa channel, and the combination of each of the four object specific channels as OD^4 channel.

6.4 Experimental Results

This section presents our experimental results in detail. All the experiments have been performed on the hollywood2 dataset [47] using the clean training dataset. The dataset is comprised of video sequences collected from 69 different Hollywood movies. There are 12 action classes in total: answering the phone, driving a car, eating, fighting, getting out of a car, hand shaking, hugging, kissing, running, sitting down, sitting up, and standing up (see Figure 5.2). In total, there are 1707 action samples divided into a training set (823 sequences) and a test set (884 sequences). Train and test sequences are collected from different movies. For performance evaluation, we report the average precisions (APs) for individual classes as well as the mean average precision (mAP) over all the action classes.

6.4.1 Baseline Performance

To get a baseline, we performed experiments with (i) the standard BoF method, and (ii) STG24 channel; using the K-Means as well as ERC-Forest generated visual dictionaries. Table 6.1 compares their mean average precisions. It turns out the STG24 channel improves upon the standard BoF approach (consistent with the findings in [36]), and the two methods perform better (about 1%) with the ERC-Forest generated dictionary. Therefore, from here on, we will only present the results obtained with the ERC-Forest dictionary.

**We use object detectors trained by the authors on VOC2008 dataset.
6.4. EXPERIMENTAL RESULTS

Table 6.2: Overall performance by different channels individually as well as in combination with other complementary channels.

<table>
<thead>
<tr>
<th>Channels</th>
<th>Performance (mean AP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS8</td>
<td>50.39%</td>
</tr>
<tr>
<td>UB</td>
<td>49.26%</td>
</tr>
<tr>
<td>OD</td>
<td>49.89%</td>
</tr>
<tr>
<td>AD-class (specific)</td>
<td>52.77%</td>
</tr>
<tr>
<td>STG24+MS8</td>
<td>53.20%</td>
</tr>
<tr>
<td>STG24+UB</td>
<td>53.18%</td>
</tr>
<tr>
<td>STG24+OD</td>
<td>52.97%</td>
</tr>
<tr>
<td>STG24+AD-class (specific)</td>
<td>55.72%</td>
</tr>
<tr>
<td>STG24 + MS8 + AD-class + UB + OD</td>
<td>55.33%</td>
</tr>
</tbody>
</table>

6.4.2 Improvements with Channel Combination

The performance by STG24 channel (51.83%) will serve as the baseline result here. Table 6.2 (1st portion) reports the results for the new channels (introduced in Section 6.3), with AD-class channel having the highest mAP (52.77%), whereas UB, the lowest mAP (49.26%). It is important to note, however, that the new channels do not improve upon the STG24 channel themselves (except AD-class channel for which we get about 1% improvement). But when the individual channels are combined with the STG24 channel, they not only improve upon their individual performances but also improve the baseline result up to 55.72% (previously 51.83%). This can be described by the fact that none of the new channels captures the spatial distribution of features as does the STG24 channel, and thus the combination with the STG24 channel results in significant improvement to the performance of the new channels. When we combine all the four new channels with the STG24 channel, we get 55.33% mAP, which is slightly lower than the best combination of STG24 + AD-class (55.72%). This performance decrease highlights the need for some more sophisticated kind of kernel combination techniques.

In Table 6.3, we present the per-class average precisions corresponding to the baseline channels as well as the best performing new channels and their combinations. The important thing to note is that we improve the performance of eleven out of twelve action classes (APs in bold in the last two columns), when we combine our new channels with the baseline STG24 channel. Moreover, though the mean AP performance by the final channel combination (55.33%) is slightly lower than that by the STG24 + AD-class channel (55.72%), yet it achieves the best results for seven action classes (APs in bold in the last column). It is important to note that the small decrease in the mean AP is particularly because of the significantly lower performance for the HandShake class (38.41%).
## 6.5 Conclusion and Future Works

We have presented an extension to the standard BoF approach for classifying human actions in realistic videos. Our main idea is to segment videos into semantically meaningful regions (both spatially and temporally) and then compute histogram of local features for each region separately. As we have shown experimentally, this separation helps to get significant improvement over our strong baseline by disambiguating histograms of local features. Our framework also enables introduction of additional supervision into BoF action classification in the form of region detectors that could be trained on related tasks.

While we have shown significant improvement with the proposed method, we believe even higher improvement could be obtained by using a more appropriate procedure for combining different channels. Our main focus in this chapter has been to demonstrate a notion of semantic level video segmentation and its advantage for action recognition. We plan to optimize channel combination and better adapt our method for each particular action class in the future work using frameworks such as Multiple Kernel Learning [2, 28].

---

### Table 6.3: Per-class AP performance by different channels/channel-combinations.

<table>
<thead>
<tr>
<th>Channels</th>
<th>BoF</th>
<th>STG24</th>
<th>AD-class</th>
<th>STG24 + AD-class</th>
<th>STG24 + MS8 + AD-class + UB + OD</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean AP</td>
<td>48.55%</td>
<td>51.83%</td>
<td>52.77%</td>
<td>55.72%</td>
<td>55.33%</td>
</tr>
<tr>
<td>AnswerPhone</td>
<td>15.71%</td>
<td>25.87%</td>
<td>20.75%</td>
<td>26.32%</td>
<td>24.77%</td>
</tr>
<tr>
<td>DriveCar</td>
<td>87.61%</td>
<td>85.91%</td>
<td>86.87%</td>
<td>86.48%</td>
<td>88.11%</td>
</tr>
<tr>
<td>Eat</td>
<td>54.77%</td>
<td>56.39%</td>
<td>57.38%</td>
<td>59.19%</td>
<td>61.42%</td>
</tr>
<tr>
<td>FightPerson</td>
<td>73.90%</td>
<td>74.93%</td>
<td>75.73%</td>
<td>76.21%</td>
<td>76.47%</td>
</tr>
<tr>
<td>GetOutCar</td>
<td>33.35%</td>
<td>44.02%</td>
<td>38.26%</td>
<td>45.71%</td>
<td>47.42%</td>
</tr>
<tr>
<td>HandShake</td>
<td>19.99%</td>
<td>29.68%</td>
<td>45.71%</td>
<td>49.73%</td>
<td>38.41%</td>
</tr>
<tr>
<td>HugPerson</td>
<td>37.80%</td>
<td>46.08%</td>
<td>40.75%</td>
<td>45.41%</td>
<td>44.58%</td>
</tr>
<tr>
<td>Kiss</td>
<td>52.12%</td>
<td>54.96%</td>
<td>56.00%</td>
<td>58.96%</td>
<td>61.47%</td>
</tr>
<tr>
<td>Run</td>
<td>71.13%</td>
<td>69.40%</td>
<td>73.18%</td>
<td>71.97%</td>
<td>74.31%</td>
</tr>
<tr>
<td>SitDown</td>
<td>59.01%</td>
<td>58.89%</td>
<td>59.59%</td>
<td>62.43%</td>
<td>61.26%</td>
</tr>
<tr>
<td>SitUp</td>
<td>23.90%</td>
<td>18.40%</td>
<td>24.06%</td>
<td>27.52%</td>
<td>25.50%</td>
</tr>
<tr>
<td>StandUp</td>
<td>53.30%</td>
<td>57.41%</td>
<td>54.94%</td>
<td>58.76%</td>
<td>60.41%</td>
</tr>
</tbody>
</table>
Chapter 7

Conclusion

In this thesis we investigated the problem of ambiguity of local features in visual recognition. We studied this problem in details on video classification task as well as image classification task.

We approached the problem of disambiguating local features from a dichotomy of perspectives which are discussed in the following. In one view we try to incorporate spatial information into local features and BoF paradigm. In that regard we made an empirical study on importance of localized grids in disambiguating BoFs. In order to make a valid evaluation of the amount of improvement one could get by accommodating BoF with respect to location of objects (incorporating spatial information) we use information of object/actor bounding box in the experiments that correspond to this view (Chapter 1 and 3). Our experiments proved that when spatial grids are localized dynamically w.r.t. a centric object (as opposed to static spatial grids) one could get huge improvement (15%) over image classification task. Although this improvement is considerably less on the task of human actions, but, two important factors should not be ignored. Firstly, our action recognition task is very challenging and because of huge amount of viewpoint, scale, appearance, etc. variations in samples of the database it is comparably hard to apply a solution that works good for all of the action classes. Secondly, using bounding box annotation, even if is not helpful by itself but it provides potentials for utilizing more information. As an example, adding spatiotemporal grids (even static grids), to bounding box of actors results in a remarkable boost-up in performance.

In the second view we proposed to add global level attributes to local features and we showed that this helps to disambiguate similar local features extracted from incoherent regions of image/video. We showed that the information provided by different attributes is complementary and combining them gives rise to significant and consistent improvement of baseline approach in a wide variety of problems including image classification, human action recognition, and image retrieval. Despite the complementarity of attributed local features combining information provided by different attributes is a tricky problem. Although we could get remarkable improvement by a naive combination scheme, being multiplying kernel matrices with...
equal weights, we found it very useful to establish a smarter framework for kernel combination where different kernels are combined for each category separately.

7.1 Future Works

Our observations showed that finding a mechanism for combining multiple kernels in an optimal way is considerably helpful in our attributed local features. Once we have such a mechanism one could think of addition of more attributes. Following comes a list of suggested attributes:

- For images
  - Depth information
  - Object boundaries
  - Texture properties

- For videos
  - Skin segmentation
  - Camera motion estimation
  - Face detection

Another interesting direction for further investigation is to integrate off-the-shelf object/action detectors with our Object Centered Grids instead of using annotated bounding boxes. For the case of human actions it is also interesting to study possibilities for adding temporal information in a dynamic way (e.g. using HMMs).

\[1\] Each kernel corresponds to a certain type of attributed local features. Therefore, number of kernels is equal to the number of attributes that one uses.
Bibliography


