Global Self-Similarity and Saliency Measures Based on Sparse Representations for Classification of Objects and Spatio-temporal Sequences.

A DISSERTATION

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Guruprasad Somasundaram

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Doctor of Philosophy

Nikolaos Papanikolopoulos

December, 2012

Acknowledgements

I owe my gratitude to several people who have been the guiding light in shaping my educational career leading up to my PhD. I cherish my association with the University of Minnesota and the Department of Computer Science.

First and foremost, I sincerely thank Prof. Nikolaos Papanikolopoulos for his guidance, vision and his belief in me, which were the most influential factors in shaping my PhD. I would like to thank him for the mentorship and friendship he provided to me. I thank him for sticking by my side through the difficult times and providing the trust and encouragement I needed. He technically inspired me at several stages of my PhD. He introduced me to the field of Computer Vision and created the opportunity to work in very interesting projects right from the beginning. He had the vision to see the potential of a nascent idea and helped convert it into a Thesis.

Next, I would like to thank Dr. Vassilios Morellas who joined our lab from Honeywell and helped every one of us to step it up a notch. He introduced us to the realm of sparsity through an extended review of the IMA workshop. He always encouraged us to take on

interesting technical problems which can potentially lead to many applications. He was instrumental in establishing key collaborations that led us to solve many problems and a lot of technical papers and journals came out of them.

I would like to thank Prof. Arindam Banerjee who introduced many of the students in our lab to the field of Machine Learning and Data Mining. His methods of teaching and the excitement he sparked in us are unforgettable experiences. I had the opportunity to collaborate with him and Dr. Alexander Truskinovskiy which I relished.

I would like to thank Prof. Thomas Smith for the opportunity to work with him on Pedestrian Crosswalk Warning Efficacy Analysis project. I learned a lot from working with him. That project eventually led to many transportation related work with which I got involved. Not only did I learn to appreciate the technical aspects of solving the real world problems, but I could also see the impact it could have on society. This has been influential in deciding my future career path. I also thank him for providing valuable reviews for many of reports, papers and my PhD thesis.

I thank Prof.Volkan Isler for agreeing to be the chair of my PhD committee. His comments during the preliminary oral examination made me question some of the choices I had made in my approach and helped me solidify the basis more. He provided me the opportunity to give a lecture on object recognition to his Computer Vision class and I thank him for that. I also thank him for taking the time to review my thesis progress well ahead of the final defense. I would like to thank Dr. Ravishankar Sivalingam and Dr. Anoop Cherian who have been very good friends and inspirational colleagues. I

would also like to thank William Toczyski for all the extended philosophical and technical inputs to my thesis. I would like to thank Dr. Ajay Joshi, Dr. Evan Ribnick, Dr. Stefan Atev and many others whose association has inspired me in many ways. I would like to thank Dr. David Tolliver and Dr. Niels Haering for all the inspiration, input, and motivation they provided me during the internship at Object Video.

"Shared grief is half the sorrow, but happiness when shared, is doubled": When I hear that adage, I can only think of my best friends Dr. Sriram Doraiswamy, Niranjana Sriram and Rajapandiyan Asaithambi . I would also like to thank Karthik Krishnakumar, Karthik Nagasubramaniam and Karthikeyan Shanmugam for their friendship and moral support over the years.

I would like to express my love, gratitude and respect towards my parents: Mr. Somasundaram and Mrs. Rathnakumari and my brother Shyam for everything they have done for me. I thank my wife Anuradha for being by my side during the most important part of my life. I thank her for her love, patience, care and attention, and her unshakeable trust in me to succeed.

Dedication

To those who held me up over the years

Abstract

Extracting the truly salient regions in images is critical for many computer vision applications. Salient regions are considered the most informative regions of an image. Traditionally these salient regions have always been considered as local phenomena in which the salient regions stand out as local extrema with respect to their immediate neighbors. We introduce a novel global saliency metric based on sparse representation in which the regions that are most dissimilar with respect to the entire image are deemed salient. We examine our definition of saliency from the theoretical stand point of sparse representation and minimum description length. Encouraged by the efficacy of our method in modeling foreground objects, we propose two classification methods for recognizing objects in images. First, we introduce two novel global self-similarity descriptors for object representation which can directly be used in any classification framework. Next, we use our salient feature detection approach with conventional region descriptors in a bag-of-features framework. Experimentally we show that our feature detection method enhances the bag-of-features framework. Finally, we extend our salient bag-of-features approach to the spatio-temporal domain for use with three-dimensional dense descriptors. We apply this method successfully to video sequences involving human actions. We obtain state-of-the-art recognition rates in three distinct datasets involving sports and movie actions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human perception can be thought of as responses to visual cues presented in the image. In other words, there are regions in an image that attract visual attention. In Figure 1.1 the parts that attract visual attention are unclear and often subjective. Some of us could be interested in the snow-clad mountains in the background whereas some could be interested in the group of people who are relaxing. Cues that drive visual attention are considered top-down triggers (saliency). Saliency can be defined as the state or quality of an object to stand out relative to its neighbors [3]. A computer algorithm can be trained to determine the most visually salient regions in a top-down manner by recognizing the various objects present in the image first and then inferring if they can trigger visual attention. Alternatively, saliency can be inferred in a bottom-up manner completely agnostic to the objects present in the image. In such a scenario, saliency can act as a feature space in which such an observation can be made—i.e. the uniqueness can

be inferred. Although not in the context of images specifically, Smith *et al.* talk about the human performance dependence on design features of the performance environment and provide empirical support for the context specificity of important features [4]. We are interested in saliencies that exist in image patches (small regions of pre-determined sizes) and can be computed at low level and then any additional inferences could be made at higher levels. We can then demonstrate how saliency is used to create visual models for various object classes in order to carry out some computer vision tasks. The relationship between visual attention and the image saliency scores we determine is interesting to investigate. A brief discussion is presented in Chapter 4. We consider visual attention prediction as a neuroscience application of saliency. In the field of computer vision, saliency has been relevant for feature detection, foreground prediction, and compression. An overview of how saliency can be utilized in different fields is depicted in Figure 1.2. We address visual attention, foreground prediction, and feature detection as applications of our model of saliency.

It will be interesting to look at what can be considered as a saliency. In Figure 1.3 we can see a common sedan. However, it is rare to spot a sports car, and most features of a sports car (see Figure 1.4) may seem uncommon. For instance the spoiler can be uncommon if we have not seen many with a spoiler before. Other saliencies like rust patches, dents, paint blots, missing car parts leave behind visual cues that provide a unique identity of the object to the human eye (see Figures 1.5 and 1.6). Saliencies like these tend to be defined subjective to that specific car irrespective of what we have

Figure 1.1: Image to illustrate the subjective nature of visual attention.

seen before. For instance, a black paint patch in a white car is an outlier for that white car, but black paint is not salient for a black car. Hence we might fail to capture these saliencies if we model our defects as being outliers across the entire sample space of cars. Saliencies defined in such a way with respect to the same object can be strategically used to model the object for recognition.

One way to go about detecting saliencies is to use a classification approach. If we attempt to explicitly classify salient vs. non-salient patches, the problem becomes dependent on training information and less bottom-up in nature. It also becomes a highly supervised problem. It is possible that the classifier memorizes the training data and becomes less flexible when presented with unprecedented saliencies. If we divert our attention to only the salient regions in the image, then we can argue that any addition to this set of salient regions or removal from this set can be striking. Such an observation

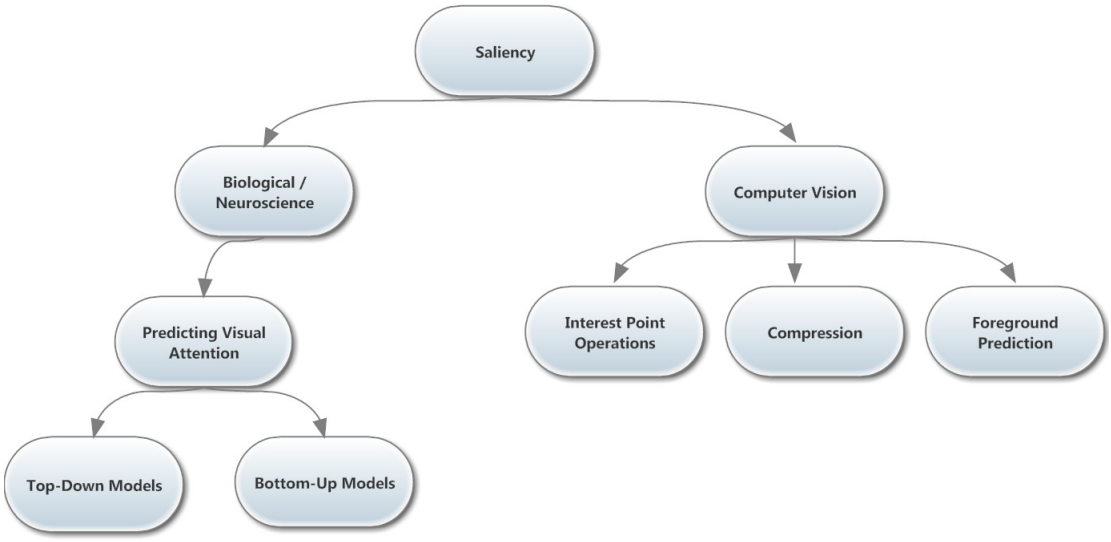

Figure 1.2: Applications of saliency in neuroscience and computer vision.

can be made if a context were to be defined for this set of salient image regions. This context can be spatial, temporal or both. For the case of still images of cars, we only need to consider spatial context of these saliencies. In other words, the image regions around the door area of a car are fairly uniform (non-salient) such that any existence of highly varying (salient) texture in those areas may be considered abnormal.

To get an idea what our saliency detection produces for an image of a car we illustrate commonly observed saliencies in Figures 1.7 and 1.8. The details of how these saliency scores are computed will be presented in Chapter 3. We notice how the wheel and window areas of the cars have more saliency than the rest of the car. We attribute this to the existence of unique textures (patterns) in these areas. We also notice that in the pizza delivery car (Figure 1.8) when the hub caps are missing the wheel areas are not

Figure 1.3: A regular 4-door sedan.

Figure 1.4: A sports coupe.

salient. This encourages the idea of using higher level inference to detect missing parts in vehicles using saliency as a feature space.

1.1 Contributions

We can summarize the main contributions of this thesis as follows. We propose a novel global saliency detection method based on sparse representations. We provide theoretical motivation as to how our approach is justified based on the theory of Kolmogorov

Figure 1.5: An old-fashioned notch back with rust patches.

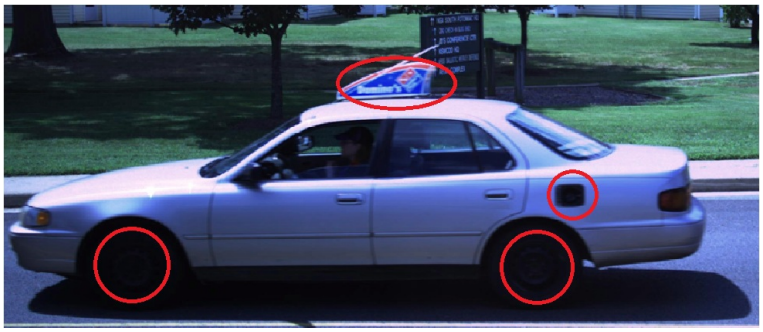

Figure 1.6: A pizza delivery car with missing gas-lid and hub caps.

complexity and description length. We also experimentally verify the efficacy of our saliency detector on foreground detection as well as visual attention datasets. We provide two ways of using our saliency detector for the problem of object classification. We propose the use of saliency weighted features in a traditional bag-of-features [5] approach with the use of popular image descriptors like PHOG [6] and region covariance [7]. We also provide two image global self-similarity descriptors based on the computed saliencies of each image region within an object boundary to describe an object for classification [8]. We finally show how our salient feature detection can be extended to the problem of action classification in video sequences and show its relevance to the

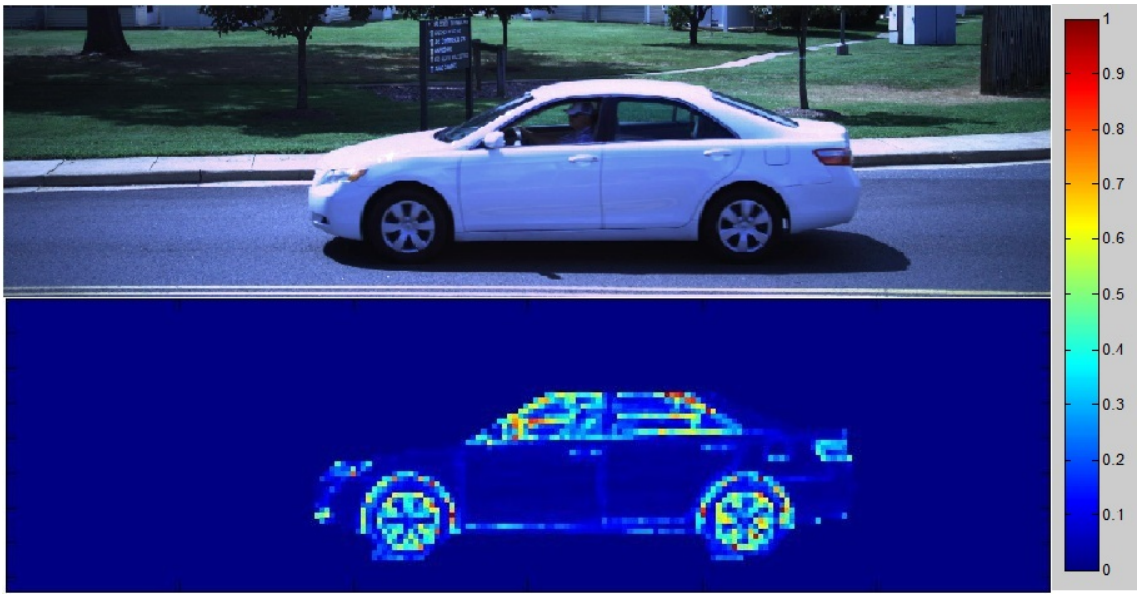

Figure 1.7: A regular sedan and corresponding saliency scores.

spatio-temporal domain [9].

1.2 Organization of the Thesis

In the next Chapter (Chapter 2) we will discuss the relevant work in a literature survey. We will explore the different types of low level features broadly. The purpose of this chapter is to provide relevance to our approach of feature computation. We discuss prior work for each application individually in each chapter. This is followed by detailed theoretical motivation and formulation of our approach in Chapter 3. We then move on to our three main applications. First, we demonstrate the efficacy of our approach in delineating foreground objects in single images as well as predicting visual attention

Figure 1.8: The pizza delivery car and corresponding saliency scores.

in Chapter 4. Next, we describe our methods for object classification in Chapter 5 and Chapter 6. We detail our methods for computing global efficient self-similarity descriptors for object classification. We provide experimental analysis using the Pascal VOC 2007 object dataset. Then we describe the salient bag-of-features approach. We show the performance of this method in real traffic video data for the classification of bicyclists vs. pedestrians. We show how the use of salient features improves traditional methods of object classification by combining different features in a Naive Bayes method as well as a concatenated histogram approach. Finally we extend our analysis to spatio-temporal sequences involving 3 datasets of human actions in Chapter 7. We use a similar saliency weighted bag-of-features approach with three-dimensional descriptors and show performance competitive with the state-of-the-art.

Chapter 2

Related Work

From a computer vision stand-point, saliency detection is akin to feature detection. In this chapter we provide relevance to our contributions by placing our approach comparatively with other feature detection methods in the literature. This is important because objects (object classes) can be represented using many types of features. These features can be computed from the entire image, interesting regions or interesting points. Many researchers have come up with descriptors for point features and patch features which are effective for learning object models and matching. There are useful features in every level of abstraction of an image. The basic feature from an image is the pixel intensity or the RGB tuple. There are still many algorithms that produce state-of-the-art performance with only RGB or gray scale values [10]. For specific applications it is often useful to investigate different color spaces. The HSV color space is considered to be close to the human perception model [11] and it is considered to be more useful in

computer graphics. There are other color spaces like HSV (hue, saturation and value), CMYK (cyan, magenta, yellow and key), CIELAB (International Commission on Illumination, luminance, a, and b color space) etc., which approximate human perception model closely [12] and have found uses in computer vision for classification applications. Specifically in [13] the use of HSV color histograms for image classification in a manner similar to document classification using Latent Semantic Indexing ([14]) is discussed.

The next level of feature detection is probably edge detection (first and second order differences across adjacent pixels). Edges are boundaries or discontinuities in image intensity values. There are many edge detectors like Sobel, Prewitt, Roberts' Cross, Canny etc., which have varying levels of complexity and performance [15]. Corners are points in the image which are intersections of two edges and are the next level of features. Like edge detectors there are many corner detectors. Harris corner detector is perhaps the first robust corner detector and is based on the weighted sum of squared differences (SSD) between image patches along the different directions [16]. Shi and Tomasi [17] provided an improved corner detector based on the Harris corner detector by measure invariance to affine transformations. In [18] an extended discussion of advanced scale and affine adapted corner detectors is presented.

Shapes can be considered as a collection of contours which are in turn a collection of edges and therefore make the next level of feature description. Other local edge and corner based features can also be considered in this level. Shape contexts [19] have been demonstrated to be a viable choice for matching and recognizing digits, letters, and

3D objects. SIFT (scale invariant feature transform) [20] is one of the most effective interest point detectors. SIFT provides more robust features than corners by finding the extrema in the scale-space of the image.

Some global image features are more suitable under certain circumstances. PHOG (pyramidal histogram of oriented gradients) provides statistical information of edge directions which is understood to be a good indication of object and shape representation in images [6, 21]. Good results for human detection in images have been made possible using these features. It is to be noted that these features depend on the quality of the images and presence of sufficient gradient information. GIST (gist of an image or scene) features provide a much more holistic multi-scale description of "scenes" and provide good classification accuracy in large scale datasets [22].

Saliency detection has been approached differently before by Kadir and Brady [23]. They measure saliency at multiple scales as patches that have high entropy. The entropy is measured by the probability density function (pdf) of local patch variations. They provide a parametric approach to filter patches with low entropy and to measure local support. In contrast to this approach, our notion of saliency differs in definition. For instance the local entropy of a patch being high may not make it salient by itself. But the pattern can be unique or less like other patterns in the entire image providing a different and more global definition to saliency. Saliency detection for the purpose of feature computation has been approached predominantly as a local phenomenon. In this thesis we propose detecting saliencies globally with respect to the entire image and

we determine its usefulness in various applications.

In summary the features discussed in this chapter vary in the detail level they extract from images (See Figure 2.1). The saliency detection algorithm presented in this thesis extracts distinct patterns which are least like other patterns in the image. This idea will be elaborated more in Chapter 3. Such an approach can be thought of as a discriminant process. This is because we try to measure the "likelihood" of a given patch with respect to the observation made about the remainder of the image or data ensemble. Even though in our approach we do not measure this likelihood directly, our saliency measure depicts the description length of each region which can be related to the probability of the region. Our method shares some similarities with other approaches which either detect saliencies in a global sense or the manner in which saliency is measured as a discriminant process. In Figure 2.2, we observe a global discriminant approach to detecting saliencies described in [24]. In this approach the Euclidean distance of LAB color space values of each pixel from the average LAB color space value of a smoothed version of the image is directly considered as saliency. The authors of this work predominantly employ this approach for foreground object detection and segmentation. In Figure 2.3, the spectral residual approach to measuring saliency is shown. In this approach, the salient regions are the residual regions which are left over when a smoothed version of the image is subtracted from the original image. The resulting high frequency regions are considered salient [25]. Bruce *et al.* use a local center-surround approach [26] but use a discriminant way of determining if the center patch is salient

based on the statistics of basis coefficients obtained via independent component analysis (ICA). This process is shown in Figure 2.4. Gao and Vasconcelos also employ a similar approach [27] for determining local center-surround saliency based on the likelihood of Gabor wavelet coefficients modeled using a Gaussian density function.

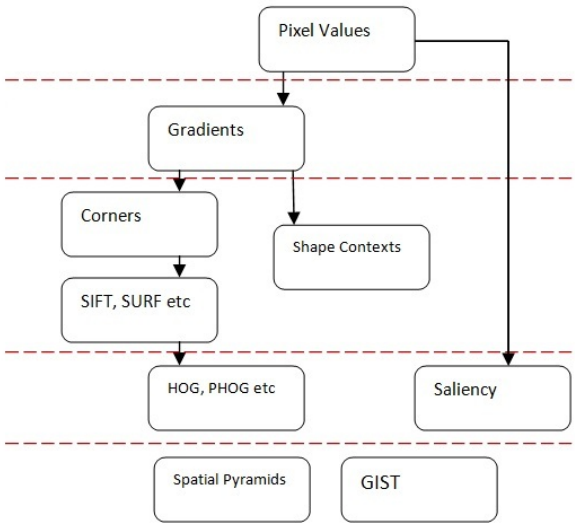

Figure 2.1: Hierarchy of image features. Higher level features are more holistic than the low level features which represent local saliencies.

2.1 Summary

In this chapter, we described relevant work for feature computation and saliency detection in a general sense. The saliency detection method as a feature detector, or a foreground detection method for various applications are described with more relevant

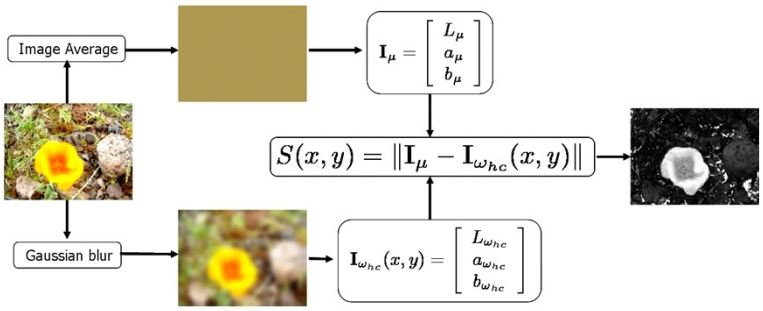

Figure 2.2: The frequency tuned approach to saliency detection. Saliencies are based on Euclidean distances in the LAB color space.

work specific to the application in the chapters to follow.

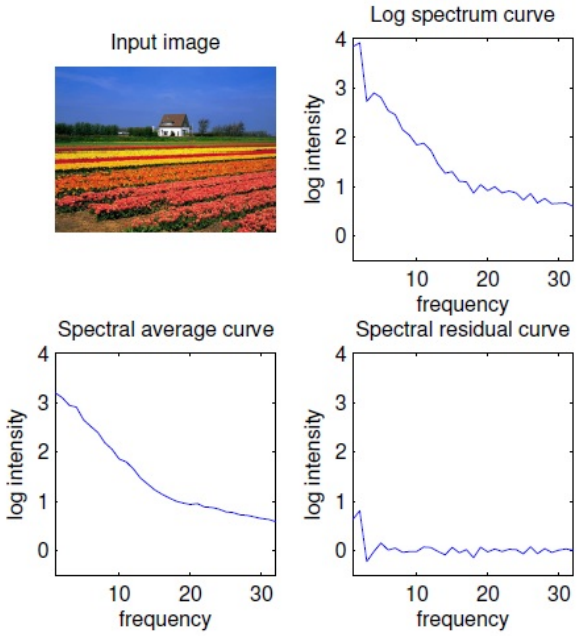

Figure 2.3: The spectral residual approach to measuring saliency. High frequency elements left after removing the average spectrum are considered salient.

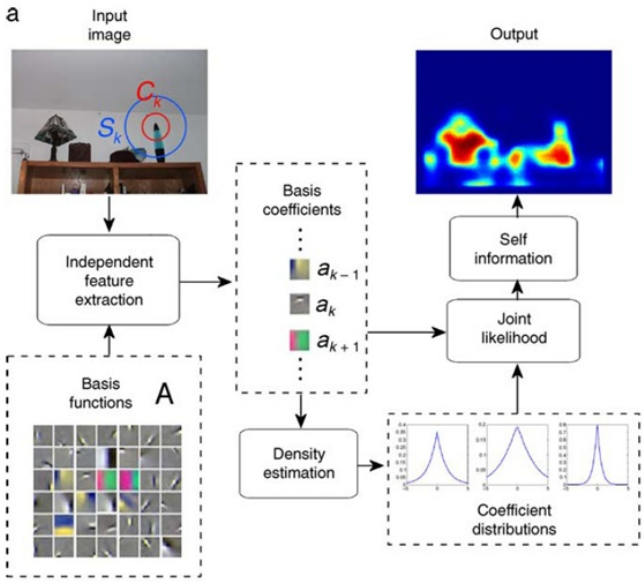

Figure 2.4: A bottom-up center surround saliency based on densities of ICA basis coefficients.

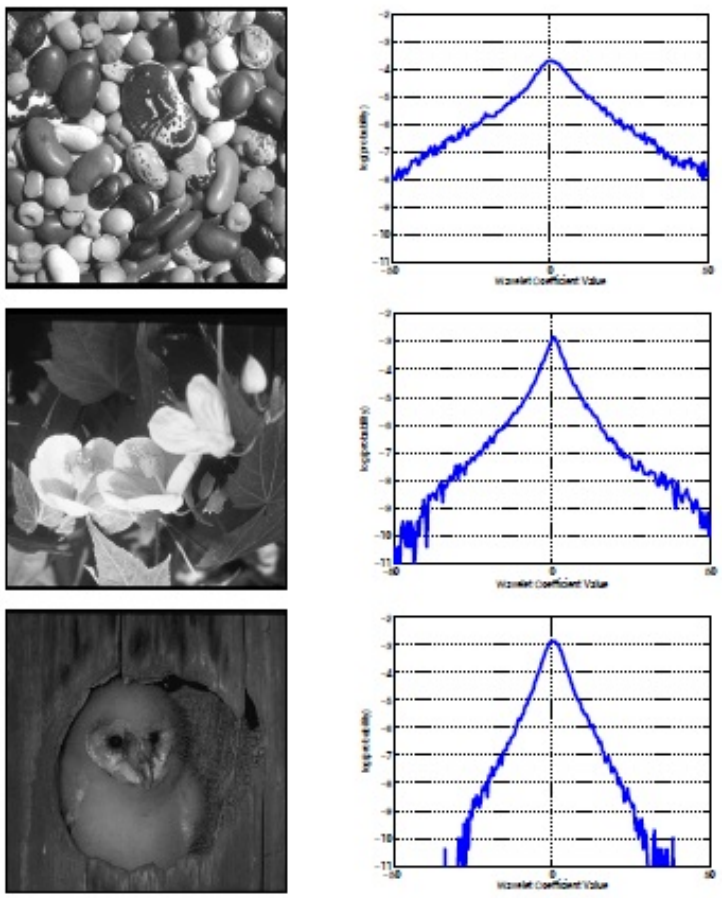

Figure 2.5: A bottom-up center surround saliency based on densities of Gabor wavelet coefficients. Here the distributions of band-pass filtered Gabor coefficients for some natural images are shown which are used to determine the saliency as a likelihood function.

Chapter 3

Theory and Algorithms

In this chapter, we motivate our approach from a theoretical stand-point. We draw motivation from the theory of Kolmogorov complexity and entropy. We aim to determine the most informative regions in an image or video sequence as defined by its description length. The more complex (or longer) the description length of a region (patch) is the more informative or complex is the patch. This chapter is divided into three parts: theoretical motivation, problem formulation, and the algorithmic approach.

3.1 Motivation

Cohen *et al.* [28] present a detailed discussion of compressed sensing and best approximations for signals. Let us assume a signal space $\mathbf{X}$ which is a compact set. For example the set of all vectors which have a bounded ℓ_2 norm form a compact set. Given a compact set, as shown in Figure 3.1, the number of ϵ balls required to get the optimal

cover N_ϵ defines the complexity of the space $\mathbf{X}$.

Then the Kolmogorov entropy of the signal space $\mathbf{X}$ is given by

$$H_\epsilon(X) = log_2 N_\epsilon(X). \tag{3.1}$$

One could think of the optimal cover of the space as the minimum number of ϵ balls (centroids) within which all the vectors in the set $\mathbf{X}$ can be represented. Since we need only $N_\epsilon(X)$ unique elements, the entropy or the number of bits required to uniquely represent each ϵ ball is given by $log_2 N_\epsilon(X)$. In practice, we cannot estimate this cover for a space of signals. However, we can attempt to learn a basis that can closely approximate this cover. Additionally, each signal in $\mathbf{X}$ does not need to be contained within one ϵ ball but rather can be represented by a combination of multiple basis vectors.

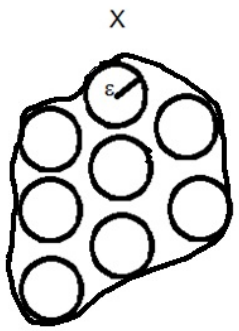

Figure 3.1: An illustration of an optimal cover within an ϵ threshold for a compact set.

Let us assume that such a basis D has been learned. Then let Σ_k be the class of

signals $\in \mathbf{X}$ that have a k sparse representation in the basis,

$$\Sigma_k := \{s = \sum_{r \in \Lambda} c_r \, d_r, |\Lambda| \leq k\} \tag{3.2}$$

where d_r and c_r are the basis vectors and their corresponding coefficients.

Not all signals will however have a perfect k-sparse representation but may be approximated with a sparse representation. The performance of the sparse representation is evaluated by the compressibility measure representing the lowest achievable distortion with the learned basis D:

$$\sigma_k(f) := \inf_{s \in \Sigma_k} \|f - s\| \tag{3.3}$$

where the norm can be any measure of distortion, although typically an ℓ_2 or ℓ_1 norm is used for reasons of ease in formulation. Under such formulations the complexity of the vector space was earlier defined by $H_\epsilon(k)$, and the complexity of each individual signal additionally is provided by its description length. In practice, if a signal can be approximated by its k-sparse representation, then its description length is k, if not it is higher than k.

The Kolmogorov complexity of each signal is given by the sum of the complexity of the model as well as complexity of the signal. For a signal that does not have approximability with a k-sparse representation, we can state it has a higher complexity given the describing basis D. Even though the complexity is an absolute quantity and practically indeterminate, we fix the size of the basis (model) and then measure the

complexity of the signal relative to the basis by way of the error incurred with an allowed description (representation) length. As mentioned earlier, we rank the patches according to their sparse representation error; hence our method is not penalized by a wrong assumption of the size of the basis D. This is because, although the actual error values may change, the rank ordering of the errors will not. Therefore for our experiments, for a patch size of $b \times b$, the vectorized patch dimensionality is b^2 and the size of the learned basis D is also fixed at b^2.

The Kolmogorov minimum description length λ [29] can be given by

$$\lambda = -log(p(x|D)) + K(D) \tag{3.4}$$

where

$$K(D) = m \times log(\sqrt{n}) \tag{3.5}$$

is the Rissanen's approximation of the Kolmogorov complexity of the model (dictionary) D. We see that lowering the description length maximizes the log-likelihood of x given the model D. We assume that the complexity of the model $K(D)$ is fixed. As mentioned previously, this is difficult to estimate in practice and is hence fixed to a known size. If the dictionary has m parameters (columns of the dictionary) and n is the dimensionality of each signal then the Rissanen's approximation for the complexity can be used to define $K(D)$. Since minimizing the description maximizes the log-likelihood of a signal, this implies that the lower the description length, the higher is the likelihood.

On the contrary, a higher description length implies a lower likelihood for the signal. A lower likelihood for the signal increases the self-information of the signal thereby making it salient.

3.2 Problem Formulation

Our objective is to identify the most salient features within a single image or video stream. To extract these features we resort to a reconstructive approach to measure highly dissimilar (salient) regions in the image. As mentioned previously, we learn a basis that represents the given image or video under the sparsity constraints. A patch from the image can be sparsely reconstructed with very few coefficients of a representative dictionary (Equation (3.6)).

$$\min_{\alpha,\mathbf{D}} \sum_{l=1}^{M} \|\mathbf{x}_l - \mathbf{D}\alpha_l\|_2^2 \;\; \text{s.t.} \;\; \|\alpha_l\|_0 \leq L. \tag{3.6}$$

Here x_l is the signal to be reconstructed with the sparsifying dictionary D using the sparse coefficients α_l which of course is regularized by the ℓ_0 norm. This is a non-convex problem and it has been approximated in the literature by replacing the ℓ_0 regularization term with a ℓ_1 regularization term (3.7).

$$\min_{\alpha,\mathbf{D}} \sum_{l=1}^{M} \|\mathbf{x}_l - \mathbf{D}\alpha_l\|_2^2 \;\; \text{s.t.} \;\; \|\alpha_l\|_1 \leq L. \tag{3.7}$$

Under such a formulation, the signal x is said to have the best reconstruction

$\mathcal{R}^*(\mathbf{x}, \mathbf{D})$, where

$$\mathcal{R}^*(\mathbf{x}, \mathbf{D}) = \|\mathbf{x} - \mathbf{D}\alpha^*(\mathbf{x}, \mathbf{D})\|_2^2. \tag{3.8}$$

Here $\alpha^*(\mathbf{x}, \mathbf{D})$ is the optimal L-sparse decomposition for the pair $(\mathbf{x}, \mathbf{D})$.

3.3 Algorithmic Approach

There are two parts to this problem. Learning this reconstructive dictionary D, and given the dictionary, how to retrieve the optimal set of sparse coefficients α^* to represent the signal x is an NP hard problem. The dictionary learning procedure can be carried out using the K-SVD (K-Singular Value Decompositions) [30] algorithm, or the MOD (method of optimal directions) [31] algorithm. A greedy orthogonal matching pursuit algorithm [32] can be used for retrieving the α^* efficiently [33], despite being suboptimal.

In effect we are trying to do blind source separation on each patch. That is we would like to represent each patch as

$$\mathbf{x} = \sum_{i=1}^{L} \alpha_i s_i \tag{3.9}$$

where s_i is the i^th source signal, α is the corresponding coefficient and L is the maximum number of sources allowed for the reconstruction.

The KSVD algorithm is shown in Figure 3.2. For more details the interested reader is directed to [10]. The procedure updates the dictionary atom and the associated α sequentially. The α are retrieved non-trivially using orthogonal matching pursuit

illustrated in Figure 3.3.

Input: $\mathbf{D} \in \mathbb{R}^{n \times k}$ (dictionary from the previous iteration); M vectors $\mathbf{x}_l \in \mathbb{R}^n$ (input data); $\alpha \in \mathbb{R}^{k \times M}$ (coefficients from the previous iteration).
Output: $\mathbf{D}$ and α.
Loop: For $j = 1 \ldots k$, update $\mathbf{d}$, j-th column of $\mathbf{D}$,
• Select the set of patches that use $\mathbf{d}$:

$$\omega \leftarrow \{l \in 1, \ldots, M \text{ s.t. } \alpha_l[j] \neq 0\}. \quad (5)$$

• For each patch $l \in \omega$, compute the residual of the decomposition of $\mathbf{x}_l$: $\mathbf{r}_l = \mathbf{x}_l - \mathbf{D}\alpha_l$.
• Compute a new atom $\mathbf{d}' \in \mathbb{R}^n$ and the associated coefficients $\beta \in \mathbb{R}^{|\omega|}$ that minimize the residual error on the selected set ω, using a truncated SVD:

$$\min_{||\mathbf{d}'||_2=1, \beta \in \mathbb{R}^{|\omega|}} \sum_{l \in \omega} ||\mathbf{r}_l + \alpha_l[j]\mathbf{d} - \beta_l \mathbf{d}'||_2^2. \quad (6)$$

• Update $\mathbf{D}$ using the new atom $\mathbf{d}'$, and replace the scalars $\alpha_l[j] \neq 0$ from Eq. (5) in α using β.

Figure 3.2: The KSVD dictionary update procedure.

The OMP algorithm selects greedily at each step the atom with the highest correlation (maximum dot product) to the current residual. Once the atom is selected, the signal is orthogonally projected back to the span of the selected atoms, the residual is recomputed, and the process is repeated. It is orthogonal in that the current residual is eliminated for computing the projection in the next step.

We observe in the KSVD procedure that the dictionary accommodates new data by changing the dictionary atoms that have the most relevance to the current data sample, by modifying the pertinent atoms. Over multiple iterations (typically 20 to 30 [10]),

ORTHOGONAL MATCHING PURSUIT

Input: Dictionary $\mathbf{D}$, signal $\underline{x}$, target sparsity K *or* target error ϵ
Output: Sparse representation $\underline{\gamma}$ such that $\underline{x} \approx \mathbf{D}\underline{\gamma}$
Init: Set $I := (\,)$, $\underline{r} := \underline{x}$, $\underline{\gamma} := \underline{0}$
while (*stopping criterion not met*) **do**
  $\hat{k} := \underset{k}{\text{Argmax}} \; |\underline{d}_k^T \underline{r}\,|$
  $I := (\,I, \hat{k}\,)$
  $\underline{\gamma}_I := (\mathbf{D}_I)^+ \underline{x}$
  $\underline{r} := \underline{x} - \mathbf{D}_I \underline{\gamma}_I$
end while

Figure 3.3: The orthogonal matching pursuit procedure.

the dictionary converges to a truly representative dictionary given a large number of training samples minimizing the average training error.

We can also make our saliency measure robust to variations in scale by convolving the original image with Gaussian kernels of varying scale. Then dictionaries corresponding to each scale are learned and the error is measured with respect to the combined dictionaries (concatenating the dictionaries of different scales). We typically generate 3 scaled versions of the image by repeatedly convolving the original image I with a Gaussian kernel $g(\sigma)$ where σ is the standard deviation of the kernel.

For measuring global self similarity across multiple scales, we propose that the dictionaries be learned for all the patches in different scales separately. They can then be

concatenated together to form a single dictionary.

$$\mathbf{D_{I_{multiscale}}} = [D_{I_\sigma}|D_{I_{\sigma_1}}|D_{I_{\sigma_2}}|D_{I_{\sigma_3}}]. \tag{3.10}$$

While performing orthogonal matching pursuit with the combined multi-scale dictionary, the different scales compete with each other in trying to represent the patch of interest. In this way, the effect of varying patch sizes is also reduced. More evidence is provided in later chapters.

Chapter 4

Foreground Detection and Visual Attention Prediction

Information extraction from single images and video sequences has monopolized recent research efforts in the computer vision community. In computer vision research, a great deal of information extraction processes are considered in the context of a single application, thus questioning their importance and limiting their wide acceptance. Salient regions of images are often very informative and help in enhanced processing of the images. In Chapter 3, we proposed a novel global saliency estimation method which is based on a sparse representation scheme. This method measures how dissimilar (salient) a local region in the image is to the rest of the image. We compare our method with other methods in this domain for the purpose of visual attention prediction on the MIT eye fixation dataset [34], as well as on the MSRA foreground detection dataset [24].

4.1 Introduction

Saliency can be defined as the state or quality of an object to stand out relative to its neighbors [3]. Saliency can act as a feature space in which distinctiveness can be inferred. Saliencies are often abstract and subjective based on what the observer is interested in an image. Despite the subjective nature of saliency, recent efforts in this area have focused on methods that attempt to quantitatively measure visual attention [24, 35, 25].

We are specifically interested in saliencies that exist in the vector space of image patches (small regions of predetermined sizes) and can be computed automatically (see Figure 6.3). Our definition of saliency is related to that given by Boiman and Irani [36] according to which we find globally salient regions. We define the patches or regions in an image that are difficult to represent sparsely using a dictionary learned from patches of the image as being salient. Detailed description of our saliency metric is provided in Chapter 3.

In [34], the authors hypothesize that most saliency computation methods are bottom-up approaches and these usually do not match with eye fixation locations based on human eye tracking. They describe a data-driven approach to predict fixation locations using a trained multi-level architecture. Even though bottom up saliency methods are not suited naturally for estimating eye fixation locations, many saliency methods in the literature quantify the performance of saliency computation using correspondence measures with these eye tracking results as ground truth. In this chapter, we similarly

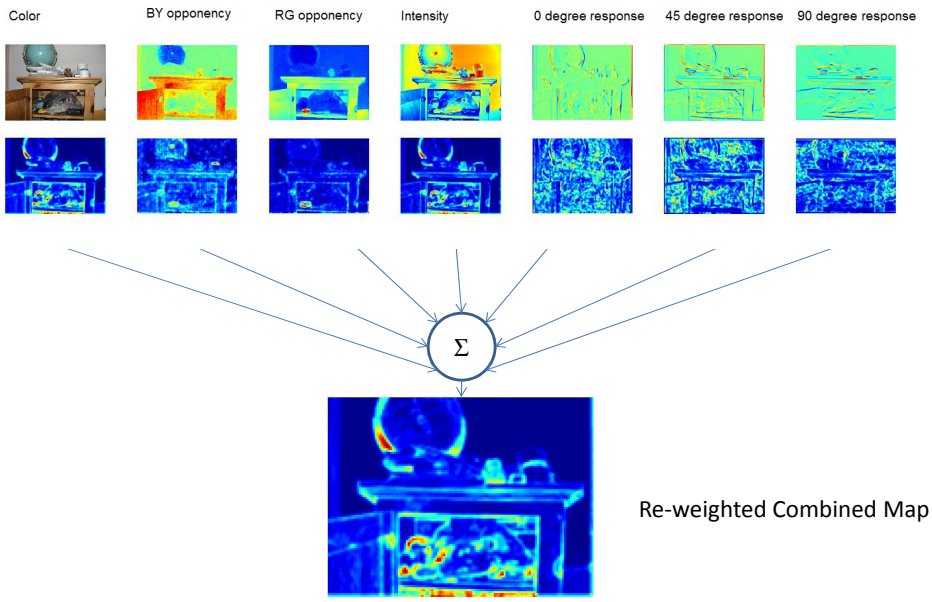

Figure 4.1: An illustration of how saliency is computed for a sample image using our method and the corresponding saliency map. This figure illustrates the procedure for predicting visual attention. For detecting foreground objects we use only the color information.

compare our approach for computing saliency with other methods in the literature using eye fixation datasets. Annotated images such as the MIT eye fixation database [34] provide a common ground for quantitative performance evaluation of saliency measures.

4.2 Prior Work

Many approaches for estimating saliency in images have been focused on being application independent. However, such a saliency measure suited for multiple applications is difficult to compute. Saliency is normally inferred in a particular feature space such as color, intensity, orientation, frequency spectrum, etc., where saliencies stand out as outliers. Itti *et al.* [37] presented one of the earlier works in saliency estimation by using a neural network to select saliencies on feature based topographical maps. Recently, Achanta *et al.* [24] proposed that salient regions can be inferred in the LAB color space as outliers. Hou and Zhang approached the problem of estimating global saliency using a spectral residual approach [25]. They analyze the residual log spectrum of an image when an average filtered version of the spectrum is removed from it. They hypothesize that salient regions in images do not conform to the general spectrum of the image. Wang and Li, while extending the spectral residual approach, argued that saliency is not just always a high variance object in a low variance background but is sometimes the other way around [35]. Further, they use Gestalt grouping principles to retain only the most contiguous salient structures in an image. Harel *et al.* [38] introduced a graph based approach for computing saliencies in images by taking into account activations

from different feature channels and then normalizing them to highlight conspicuousness. They show good precision with eye fixation data.

Following Boiman and Irani [36], we equate saliency detection with anomaly detection in the vector space of patches. They proposed that a database of patches (spatial for images and spatio-temporal for videos) which are considered normal are learned with little training data. Then patches which do not conform to the database estimated by a log-likelihood score are deemed as anomalies. The proposed method defines saliency in a similar sense. However, we use a sparse representation scheme to measure saliency which relates to identifying image patches which are hard to reconstruct sparsely given the knowledge-base (dictionary) of a single image.

Kadir and Brady [23] measure saliency at multiple scales as patches that have high entropy. The entropy is measured by the probability density function (pdf) of local patch variations. They provide a parametric approach to filter patches with low entropy and to measure local support. In contrast, the proposed approach measures saliency in a global sense as the patches which are unexpected with respect to the information available in the rest of the patches in the image.

Entropy based definitions contrast with context based approaches (e.g., Goferman et al. [39]) that define saliency based on principles found in the vision science literature. In [39], saliency is defined at the object level and in contrast to the context of objects in scene. By comparison, the proposed approach does not assume a context.

4.3 Approach

Based on the theory developed in Chapter 3, we come up with strategies for segmenting foreground objects in single images as well as for predicting visual attention. Essentially, these two problems are the same. However, segmenting the foreground object involves differentiating an object from a background. In this case, the foreground can visually be non-salient and the background can be salient in terms of what attracts visual attention. Hence, we need to use additional information while predicting visual attention. For this purpose, we rely on the findings of other efforts in the past to determine the cues for predicting visual attention. For identifying the foreground, just the intensity values of each pixel is sufficient. However, for predicting visual attention we use the color, intensity, and orientation channels of an image. For color, we use the color opponency space of RG and BY opponency as described in [27]. For the orientation channels, we convolve the original image with steerable Gaussian filters [40] of different orientations $(0, \frac{\pi}{4}, and\ \frac{\pi}{2})$ and measure the responses. The different channels for a sample image and the corresponding saliency maps are shown in Figure 6.3. For predicting visual attention, the final saliency map is obtained by taking the sum of all the different channels for each pixel and then re-normalizing so that all cues agree with the visual saliency of a pixel. Finally, we compare the saliency maps with the ground truth using the ROC area as a metric.

4.4 Experiments and Results

We performed two types of experiments: detecting foreground objects and predicting eye fixation. While detecting the foreground information, we only use simple cues such as color. However, for predicting visual attention we use color, intensity, opponency, and orientation feature spaces as shown in Figure 6.3.

4.4.1 Foreground Detection

For this analysis, we used the MSRA salient object detection dataset. Achanta *et al.* [24] provide ground truth for 1000 images from this dataset. Some sample saliency maps for this dataset are shown in Figure 4.3. The ROC areas [27] were compared for different methods. A value of 0.5 indicates a random worthless predictor, whereas a value of 1 indicates that the method is a perfect predictor. The ROC curve is obtained by plotting the true positive rates ($\frac{True\ Positives}{All\ positives}$ also known as *hit rate* or *recall*) versus the false positive rate ($\frac{False\ Positives}{All\ negatives}$ also known as *fall out*) . One such ROC curve is shown in Figure 4.2. We can obtain this by thresholding the saliency map with different threshold values and computing the false positive and true positive rates. The ROC area is considered a better performance metric than accuracy alone [41] as it is invariant to the choice of the threshold.

The average ROC areas for the different methods are shown in Table 4.1.

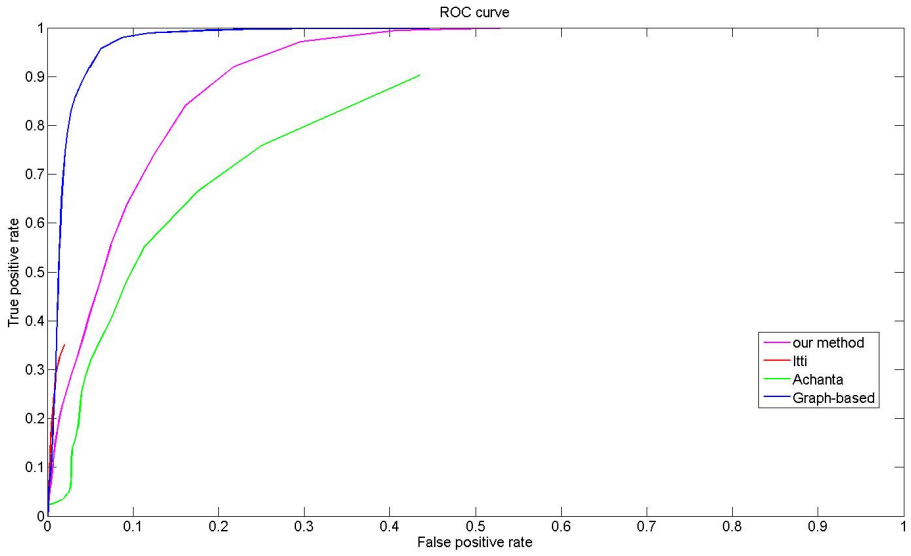

Figure 4.2: ROC plots corresponding to one image for all methods compared.

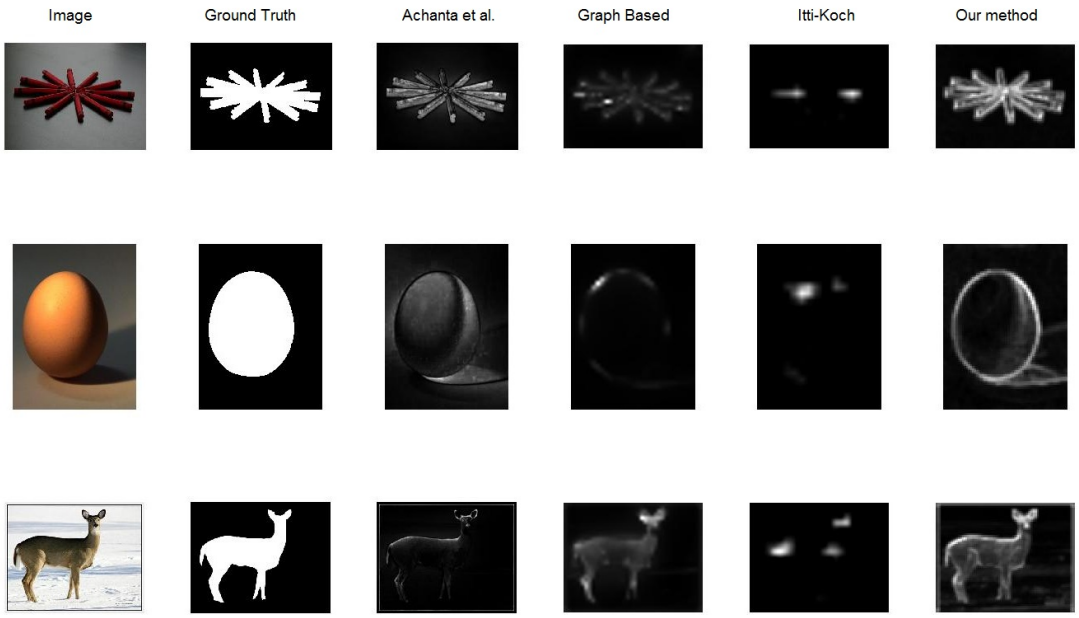

Figure 4.3: Comparison of saliency maps for some images from the MSRA dataset.

Method	Average ROC area
Itti	0.58
GBVS	0.61
Frequency Tuned	0.82
Our method	0.74

Table 4.1: Average ROC areas for different methods on the MSRA salient object dataset.

4.4.2 Predicting Visual Attention

For this experiment, we used the MIT eye fixation dataset [34]. We used their blurred saliency map obtained from eye fixation points as ground truth. With a ground truth G and a saliency map A, we can compute the precision and recall of the object segmentation as mentioned before. We used 100 randomly sampled images for training to learn the best value of M for selecting the top salient regions which yielded the best average precision and recall using the ground truth. Since eye gaze track points are sparse, choosing the top 10% salient regions yielded best results. Even though our recall was sometimes poor, we obtained the best average results overall.

The MIT eye fixation dataset is a particularly challenging dataset and is more suited for top-down saliency estimation approaches. In [34], the authors describe how data-driven methods which are trained to look for specific objects or patterns tend to have close correspondence with human eye fixation tracking. Our method performs better than the frequency-tuned saliency estimation approach [24] and the Itti-Koch saliency

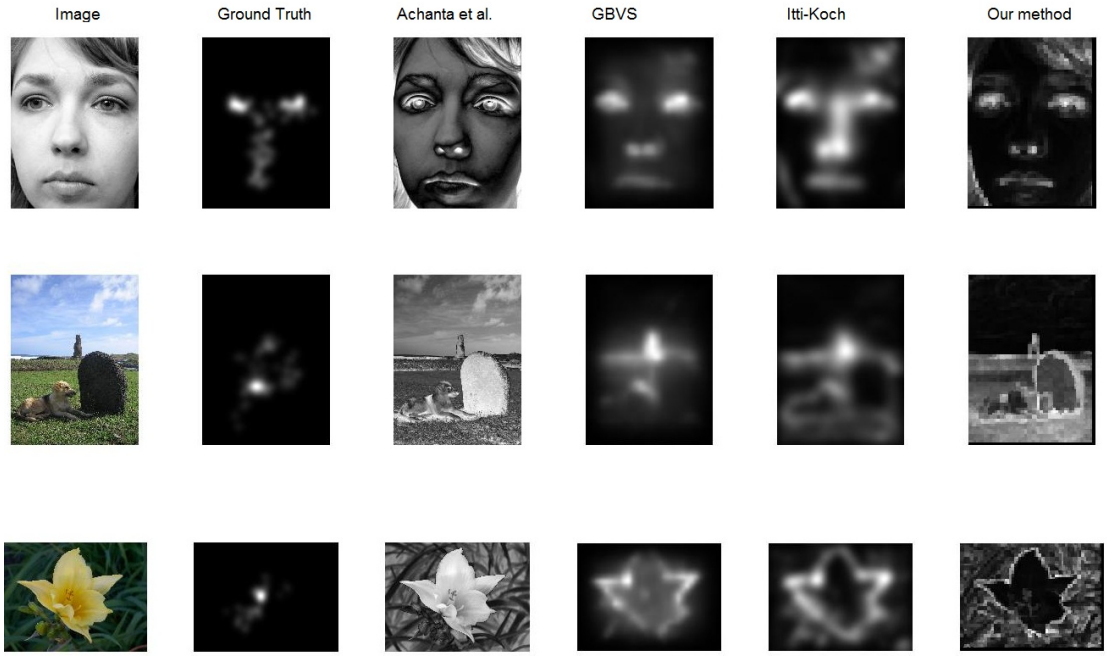

Figure 4.4: Comparison of saliency maps of different methods on the MIT eye fixation database.

map [37] which are also another bottom up approaches like our method (see Table 4.2). That is, both methods focus on estimating outliers in a feature space and do not assume the existence of a eye fixation correspondence. We notice both the graph-based visual saliency [38] computation methods provide more precision with respect to the eye tracking data. While in many cases the salient textures estimated by our method have strong correlation with eye fixation, salient textures in background or 'unimportant' regions in the images do not demand attention from a human observer. This is indicated in the saliency maps shown for some examples in Figure 4.4.

Method	Average ROC area
Itti	0.68
GBVS	0.7
Frequency Tuned	0.52
Our method	0.59

Table 4.2: Average ROC areas for different methods on the MIT eye fixation dataset.

4.5 Conclusions

We proposed a sparsity induced saliency detection approach for images. The algorithm identifies patches in images which are globally salient or least like other patches in terms of content. We have shown experimentally using two datasets that our measure is suited for predicting eye fixation as well as for detecting the foreground object. We notice that our method similar to the approach of Achanta *et al.* [24] is very good at detecting dense foreground objects with high recall because of the dense saliency maps. Itti-Koch and Graph based visual saliency methods are better for predicting eye gaze patterns and they are also more sparse. These methods however are not preferred for foreground detection. Encouraged by these results, we proceed to solve problems of object classification in images and action classification in temporal sequences.

Chapter 5

Object Classification Using Self-Similarity Descriptors

Object recognition entails extracting information about which object class(es) are present in an image. In order to enhance the performance of object recognition, reducing the redundancy in the data is absolutely essential. Prior literature [42, 43] introduced local and global self-similarity features to highlight the areas in an image which are useful for object classification and detection. We introduce an efficient self-similarity measure based on sparse representations and propose two different descriptors. Our measure of self-similarity is determined across multiple scales and is more efficient than prior work. We test our self similarity descriptor using support vector machine based classification on the PASCAL VOC 2007 database consisting of 20 object classes. Comparative results indicate performance competitive with the prior approaches of computing self-similarity

descriptors.

5.1 Introduction

The modeling approach for object classification tasks seems to follow a general trend in recent approaches. In [44, 45, 46], the problem is approached with emphasis on the individual parts of a 3D model while respecting the geometric constraints. The general philosophy of these methods is to individually detect parts, and then use the labels and geometric relationship of the detected parts to determine what class the object belongs. Instead of describing the parts individually, we form a characteristic description of the class where the characteristic elements may or may not have a semantic label, or a meaning attached. However, like the parts used in parts-based classifiers the characteristic elements are also found consistently across multiple instances of the object classes, while not requiring any effort dedicated to learning such parts. Self-similarity descriptors attempt to capture such characteristic elements of objects which are stable across changes in scale, viewpoint, etc.

Self-similarity was introduced by Shechtman and Irani [42] in order to determine patterns that can be repeatedly found in instances of objects present in different backgrounds (See Figure 5.1). Self-similarity descriptors when used along with global or local image descriptors enhance the representation power of those descriptors. In [42, 43], the authors illustrate improved classification and detection performance with the use of self-similarity descriptors. Shechtman and Irani argued that there is never a single cue

Figure 5.1: An example of our similarity map of an object. We can see that there are patterns that can be repeatedly detected across different instances of the object. Red areas indicate more dissimilarity. Images: frontal view of cars from PASCAL VOC 2007 dataset.

such as color, intensity, or orientation that can be reliably used to detect patterns of objects across multiple examples. Their idea was to determine local intensity patterns that can be found to be shared across multiple examples of objects. Deselaers and Ferrari [43] had an alternate view that these patterns exist globally (GSS) and they measured similarity of local regions with respect to the entire image. They obtained better results than using local similarity measures (LSS). Both measures are obtained using a distance measure such as the SSD (sum of squared differences). In contrast, we measure the global self-similarity score as the residue in sparse representation with a dictionary learned from the image. This dictionary is a concise representation of the image itself such that within the sparsity constraints the dictionary spans the entire image. Therefore, the reconstruction error directly represents the distance of the query

patch from the rest of the image. The reconstruction residue is in fact the complementary score of the self-similarity. The lower the error in representation, the more 'self-similar' the patch is. In other words the residual sparse reconstruction error is as informative as a self-similarity score. Also this approach has well developed and efficient methods for both the construction of the dictionaries [47] and the sparse recovery problems [10, 48, 32] making it attractive for potential real-time uses.

We discuss two representation schemes for using the self-similarity scores in a descriptive fashion: AuREC (area under reconstruction error curve) and BoREC (bag-of-reconstruction error curves). These descriptors are for higher level representation. We use the path of the reconstruction residue returned by the regularization path over the increasing levels of sparsity as a descriptor for a particular region. These reconstruction paths when quantized with respect to a codebook can be successfully matched across images. Using these descriptors we are able to achieve competitive performance in object classification. In the next section we introduce the global self-similarity measure derived from dictionary learning and sparse representations. In Section 5.3, we discuss our object classification approach based on the derived self-similarity descriptors and discuss the performance with our experimental results.

5.2 Residual Error for Measuring Self-Similarity

We will attempt to measure self-similarity of different regions within a single image. Given a set of image patches $\mathbf{X}$, a representative dictionary can be learned by minimizing

the measure in expression (5.1). The number of atoms in the dictionary is fixed equal to the dimensionality of the patches.

$$\min_{\alpha,\mathbf{D}} \sum_{i=1}^{M} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 \;\text{ s.t. }\; \|\alpha_i\|_0 \leq L. \tag{5.1}$$

Here x_i are the image patches to be reconstructed by the dictionary D using the coefficient vector α_i which has an upper-bound L on their cardinality (ℓ_0 norm). In other words any signal x_i has a limited support set in the span of the dictionary D.

That is we would like to find the closest approximation $\hat{x}$ such that $\|\hat{x} - x\|_2$ is minimized. Under such a formulation, the patch x is said to have an optimal reconstruction when the reconstruction error $\mathcal{R}$ is minimal ($\mathcal{R}^*(\mathbf{x}, \mathbf{D})$), where

$$\mathcal{R}^*(\mathbf{x}, \mathbf{D}, \alpha^*) = \|\mathbf{x} - \mathbf{D}\alpha^*(\mathbf{x}, \mathbf{D})\|_2. \tag{5.2}$$

Here $\alpha^*(\mathbf{x}, \mathbf{D})$ is the optimal L-sparse decomposition for the pair $(\mathbf{x}, \mathbf{D})$. Learning the dictionary D as well as recovering the sparse coefficients α_i for a signal x_i are both non-convex problems. However, efficient methods such as KSVD [47] can be used for the learning procedure. The sparse recovery can be done using a greedy algorithm known as the orthogonal matching pursuit [32] for which we can leverage the already existing efficient implementations [49, 50]. We denote the residual reconstruction error of a patch as the self-similarity score. That is the higher the reconstruction score the more dissimilar the patch is. Note that we do not explicitly compute the similarity of the patch with respect to every other patch in the image. Instead we just measure

the minimum ℓ_2 distance of a patch from the dictionary of patches, allowing for sparse representation. The dictionary even under the sparsity constraints of the number of atoms used guarantees a representation with upper bounds on the error. This implies that we can reliably retrieve the self-similarity score as close to the true value as possible. Even with the dictionary learning and sparse coding, this method can be faster than explicitly attempting to find the true maximum self-similarity score.

Global Self-Similarity

Given the findings of [43], we compute the global self-similarity measure and not the local self-similarity score. By learning a dictionary from all the patches in the image D_I and then measuring the reconstruction residue of each patch with respect to D_I yields the global self-similarity measure.

$$\mathcal{R}(\mathbf{x}, \mathbf{D_I}, \alpha) = \|\mathbf{x} - \mathbf{D_I}\alpha(\mathbf{x}, \mathbf{D_I})\|_2. \tag{5.3}$$

Measuring across Multiple Scales

We would like our self-similarity measures to be as robust to scale variations as possible. We account for this by constructing a Gaussian pyramid of the image. We generate 3 reduced versions of the image by repeatedly convolving the original image I with a Gaussian kernel $g(\sigma)$. Here σ is the standard deviation of the kernel.

For measuring global self similarity across multiple scales, we propose that the dictionaries be learned for all the patches in different scales separately. They can then be

concatenated together to form a single dictionary.

$$\mathbf{D_{I_{multiscale}}} = [D_{I_\sigma}|D_{I_{\sigma_1}}|D_{I_{\sigma_2}}|D_{I_{\sigma_3}}]. \tag{5.4}$$

While performing orthogonal matching pursuit with the combined multiscale dictionary, the different scales compete with each other in trying to represent the patch of interest. Therefore, the reconstruction error models the self-similarity across different scales inherently. By measuring the self-similarity across multiple scales, the effect of varying patch sizes is also reduced. We can see in Figure 5.2 that the average distribution of the self-similarity scores does not change by much for varying patch sizes. Nonetheless we still perform cross-validation analysis on different patch sizes in Section 5.3 for completeness.

5.2.1 Formation of Descriptors

We propose two novel descriptors based on the reconstruction error in Equation (7.2). The reconstruction error curve for two patches with the same final reconstruction error is shown in Figure 5.3. Even though the final reconstruction errors are the same, in the first case the error curve has a steeper slope than the second case. Therefore, instead of using just the final reconstruction error values, we use two measurements on the curve: the area under the curve and a vector quantized version of the error curve. This results in two different descriptors for each patch, one of which is a scalar value and the other a vector. These descriptors are basically a second level of description derived

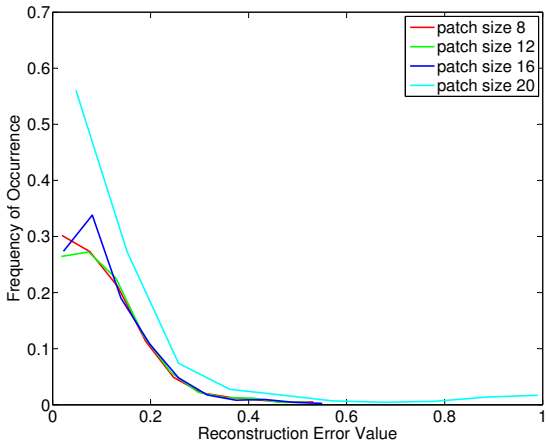

Figure 5.2: Reconstruction error distribution over 'aeroplane' images for four different patch sizes. We notice that there is not much of a difference indicating that the measure is robust to some degree of scale changes.

from the reconstruction error. The vector quantization of the error curve can be carried out using a residual error curve codebook. For instance, given that the sparsity level is 6 and that the reconstruction error $\mathcal{R} \in (0, 1)$, we can form a quantized codebook C which contains all possible permutations of the reconstruction error curves taking values $\in [0, 0.2, 0.4, 0.6, 0.8, 1]$. Because the reconstruction error is a non-increasing function with respect to the sparsity level, the number of possibilities is limited. For the sparsity level of 6 just illustrated we obtain a total of 462 unique reconstruction error vectors. This is much lower than 6^6. In general for a target sparsity level of n because of the non-increasing constraint we obtain far fewer than n^n unique error curves if the quantization factor in the error value is $\frac{1}{n-1}$.

Given the codebook C with M columns, and the image for which there are N patches whose reconstruction error curves have been retrieved, bag-of-reconstruction error curve descriptors (BoREC) can be determined by computing the histogram of the matching occurrences of the codebook columns with respect to the patches in the image. The histogram which can then be normalized serves as a M dimensional descriptor for the image. Alternatively, the area under the reconstruction error curve (AuREC) descriptor is simply the average value of the reconstruction error across all sparsity levels. This will therefore be a scalar value for each patch resulting in a N dimensional descriptor for the image. For the case of BoREC we do not need to resize the images for comparison. However, for comparing AuREC we need the value of N to be fixed which means for the image size and block size have to be fixed for all images being compared. For the BoREC we need the sparsity level to be fixed.

5.2.2 Complexity and Runtime

The time complexity of the K-SVD dictionary learning algorithm and the Orthogonal matching pursuit method are analyzed extensively in Rubinstein *et al.* [49]. Per signal (patch) complexity can be inferred from their derivation.

A Matlab implementation of the algorithm was carried out using efficient tools like the Sparse Modeling Software (SPAMS) [50]. With an efficient parallelized implementation, the run time per image of size $\approx 200 \times 200$ pixels was less than 8 seconds for AuREC descriptor and 9 seconds for BoREC descriptor given the codebook for sparsity

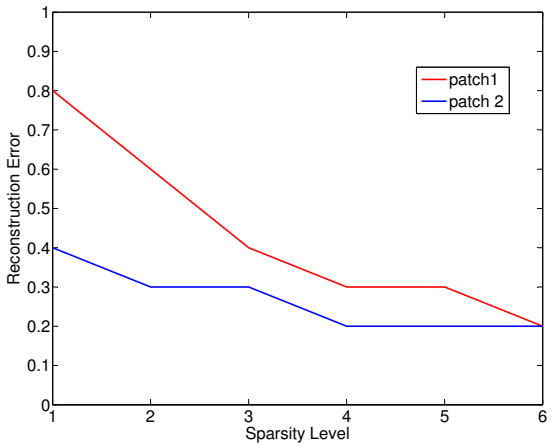

Figure 5.3: Two cases where the final reconstruction residues are the same however the manner in which they reach that value is different. These patterns can prove to be descriptive for each patch.

level 6. This is much faster compared to the time needed by the method of [43] which was around 81 seconds. The reason our method is faster is inherent to the manner in which we compute the self-similarity. We do not exhaustively attempt to determine the self-similarity score but we obtain it through a greedy optimization procedure like orthogonal matching pursuit for which efficient implementations already exist.

5.3 Classification using Self-Similarity Descriptor

For our classification experiments, we used the PASCAL VOC 2007 classification dataset [51]. This dataset contains 20 object classes: person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted

plant, sofa, TV/monitor. For each object class, there are 5 different view points: Frontal, Rear, Left, Right and Unspecified. From each class and view we sample images to form a validation set of 500 images on which we determine the best parameters for our descriptors. We perform cross-validation classification using support vector machines with linear and the radial basis function kernels. The parameters evaluated are block size, image size (for AuREC), sparsity level (for BoREC) and SVM parameters (cost parameter and gamma parameter of the rbf kernel) (see Table 5.1). We obtained best results using the BoREC descriptor for a sparsity level of 6, block size of 12×12, and for the AuREC descriptor we obtained best results with a block size of 12×12 with image size factor of 10 fixed throughout (i.e. 120×120 images). Even though in some cases AuREC performed better with more sparsity levels, we used the value of 6 because of the incurred lower dimensionality of the BoREC descriptor. For the AuREC, non-overlapping patches were considered, resulting in 100 dimensional features for the best parameters. For the BoREC descriptor overlapping patches were considered, and for sparsity level of 6, we obtained a codebook size of 462. Therefore, the BoREC also had 462 dimensions.

Using the best parameters we trained separate linear and RBF kernel SVM classifiers for the different view points of all objects and tested on the test data as performed by authors of [43]. We perform classification and report the average classification accuracy. We also combine our classifiers with a GIST descriptor based SVM classifier and report the accuracies for comparison with [42, 43]. The results are shown in Table 5.2. We

		Block Size			
	Sparsity	8	10	12	16
AuREC	6	68.1	72.3	75.12	71.12
	8	68.3	72.11	**75.43**	70.44
	10	68.3	69.22	75.06	69.03
BoREC	6	70.11	73.03	**76.12**	73.17
	8	70.08	73.01	76.08	73.16
	10	69.66	72.04	76.01	73.02

Table 5.1: Classification accuracies [%] on the validation set for different parameters. Best case results out of RBF kernel and linear kernel are reported.

obtain best performance overall with the BoREC descriptor while the AuREC is still competitive with other approaches. The reason we believe our method performs better is that we do not determine the similarity pairs based on a K-means codebook. Instead we measure a more reliable self-similarity scalar measure. Also the dictionary learned is multi-scale and represents the image more extensively under the sparsity constraints.

5.4 Conclusion

To conclude, we summarize our contributions of this chapter. We propose a novel method for measuring self-similarity based on sparse representations which is also a multi-scale measure. We also discuss two types of descriptors the AuREC and the BoREC, with which we obtain competitive classification accuracies on the PASCAL

Method	Alone	With GIST
BoLSS [42] linear SVM	25	52.9
BoLSS [42] IK-SVM	31.9	57.5
SSH [43] linear SVM	44	55.1
SSH [43] IK-SVM	45.7	59.4
AuREC linear SVM	44.8	57.2
AuREC RBF-SVM	45.3	58.8
BoREC linear SVM	58.9	63.1
BoREC RBF-SVM	**60.1**	**64.3**

Table 5.2: Classification accuracies [%] on the full PASCAL VOC 2007 data.

VOC 2007 dataset. Compared to prior approaches in the literature our method is efficient and can be easily implemented with readily available software tools for sparse representation. Even though we have discussed only classification performance in this chapter, it is clear that these descriptors can potentially be used for other tasks such as detection and appearance based tracking.

Chapter 6

Salient Bag of Features for Enhanced Classification of Traffic Objects

Object recognition algorithms often focus on determining the class of a detected object in a scene. There are usually two significant phases involved in object recognition. The first phase is the object representation phase in which the most suitable features that provide the best discriminative power under constraints such as lighting, resolution, scale, and view variations are chosen to describe the objects. The second phase is to use this representation space to develop models for each object class using discriminative classifiers. In this chapter we focus on classification of composite objects: *i.e.*, objects

which have two or more simpler classes interconnected in a complicated manner. A classic example of such a scenario is a bicyclist. A bicyclist consists of a bicycle and a human riding it. When we are faced with the task of classifying bicyclists and pedestrians, it is counter-intuitive and often hard to come up with a discriminative classifier to distinguish the two classes. We explore global image analysis based on bag of visual words enhanced by globally salient features to distinguish bicyclists from pedestrians. We also propose a unified Naive Bayes framework as well as a combined histogram feature method for combining the individual classifiers for enhanced performance.

6.1 Introduction

The problem of object classification has been approached in many different ways. However, there is very little attention in the literature to the problem of classifying composite classes such as bicyclists. The specific challenge involved in this problem is that a bicyclist is, appearance-wise, an intricate combination of a bicycle and a pedestrian thereby making it a composite class object. The proposed approach provides competitive classification results with real world data. Object classification has been addressed in many different ways, each driving its own philosophy of how an object is modeled. Despite the large number of approaches, data driven approaches are usually more structured and powerful for classifying objects. Data driven classification approaches are categorized typically based on the amount of training data provided to the model. Unsupervised classifiers have no labeled training data at all, and rely on inferring naturally existing

organization in the data in order to form clusters. This is common in data mining applications which have unlabeled example data. Unsupervised methods typically rely on kernels or basis functions or mixtures thereof to identify class structure in the data. For unsupervised image classification, denoising, and enhancement, Lee and Lewicki [52] used ICA (independent component analysis) with non-Gaussian density functions. The use of optimal binary particle swarm clustering similar to particle filtering is discussed in [53] for the purpose of unsupervised image classification.

However the need for most classification algorithms today is to learn the best possible models from a large number of labeled training data. In supervised learning, the learner is provided with labeled examples. The goal of supervised learning is to use these labeled data to come up with a decision function which will separate the data in some parametric space. The success of supervised learners depends on both the features used to represent the examples, as well as the classifier. SVMs (support vector machines) are very popular in machine learning for its high accuracy and versatility with regards to the type of data used. In [54], the use of SVMs with many different feature descriptors is detailed. Following their approach we analyzed the use of SVMs along with interest point features (SIFT and SURF) as well global image features (PHOG) for the purpose of classifying bicyclists versus pedestrians in real traffic video images [55]. The important thing to consider when using interest point features or regions is that there will be significant overlap in the features of a bicyclist and a pedestrian. This is because a bicyclist is comprised of a bicycle as well as a person in a composite manner. This motivates the

use of local image analysis on the images of bicyclists to label the individual parts of an image as to whether they belong to a person or a bicycle model. Then based on the composition of bicycle and person parts we can make a decision holistically. We employed this approach and achieved competitive performance $\approx 95\%$ [56].

By systematically combining the classification decision of these different models we obtain a very high classification performance on a set of real images obtained from real traffic scenes involving bicyclist and pedestrian traffic (bike trails, university streets etc.). We further enhance the performance using a bag of salient features classifier based on sparse representation.

Our contributions in this chapter can be outlined as follows. We develop multiple bag-of-features classifiers using local keypoint features such as SIFT, SURF, as well region descriptors such as PHOG. We also develop a salient bag-of-features classifier using our own saliency detector. We propose the use of two schemes: Naive Bayes and a combined histogram approach for the effective combination of the various classifiers. Finally we test our approach for classification and counting on real traffic video data consisting of bicyclist and pedestrian traffic under varying conditions. The performance we obtained was quite high ($\approx 95\%$ accuracy), and we compare the performances of all our approaches individually as well as in the combined state.

6.2 Prior Work

Choosing the right features for representation is often domain dependent and is very crucial for obtaining successful classification. Features can be classified broadly into interest point features, region features and shape features. Shape contexts [19] have been demonstrated to be a viable choice for matching and recognizing digits, letters, and 3D objects where shapes as well as the pose of the objects are the discriminative factors. SIFT (Scale Invariant Feature Transform) [20] is one of the most effective interest point detectors, which uses scale space extrema as interest points and provides a localized high dimensional descriptor. SURF (Speeded-Up Robust Features) were derived similarly to SIFT; however they are faster to compute and are said to be more reliable. SIFT in general is computationally expensive and in low resolution images multiple scale spaces of gradients do not carry much information. A detailed report of interest point detectors which focus on scale and affine invariance of corner detectors is given in [57]. Some global image features are more suitable under certain circumstances such as human detection. PHOG (pyramidal histogram of oriented gradients) provides statistical information of edge directions which is understood to be a good indication of object and shape representation in images [6]. It is to be noted that these features depend on the quality of the images and the presence of sufficient gradient information. Techniques that work on standardized dataset images will not necessarily guarantee good results in practice, as we need to deal with problems like poor resolution, motion blur, occlusion, etc. The problems are escalated when the images are blob images

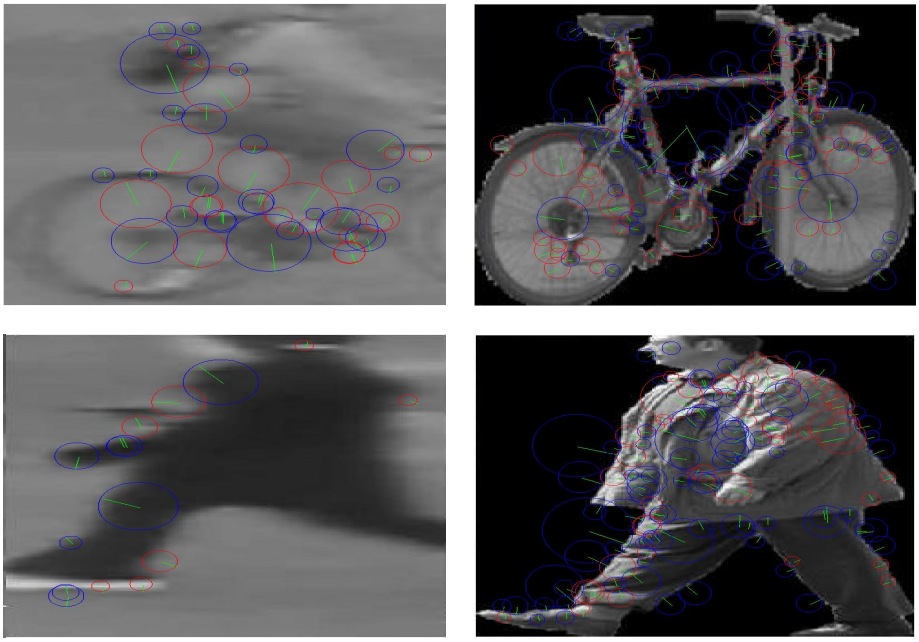

Figure 6.1: A comparison of bicyclist and pedestrian images between tracked objects (left) and samples from the Graz 02 dataset (right) [1]. We can see the difference in quality based on the density of the detected SURF [2] keypoints.

obtained from tracking (see Figure 6.1).

SVM (support vector machine) based approaches have been prominent under the supervised learning methods category [58]. Not limited to object classification, SVMs have been an attractive choice for many other classification and regression applications. SVMs have been used with many different object features with success and a detailed report is available in [54]. A similar approach which constructs different feature vocabulary sets, as well as investigates the use of probabilistic latent semantic analysis, is presented in [59]. HOG (Histogram of oriented gradients) features have been extended

to construct part models which accommodate deformations in objects and hence improve accuracy [60]. Another approach to parts modeling is to learn the spatial contexts of the different parts by learning the weights of all possible configurations of parts across different object classes [61]. In papers [62], and [63], the use of LDA (Latent Dirichlet Allocation) for object and scene classification and annotation is discussed. Most of these methods use benchmark datasets such as the Caltech, Graz 02, etc.

6.3 Approach

In this section we provide a brief description of the individual approaches we adopt for classifying bicyclists versus pedestrians based on global and local plus coarse and fine scale image analysis. We explain each individual classifier and finally we provide a Naive Bayes approach for combining the classification decisions of the individual classifiers.

6.3.1 Bag of Visual Words Classifier

The Bag-of-Visual-Words (BoVW or BoW) method for modeling objects is derived from the general principles of Natural Language Processing. The model learns "words" or meaningful features of different object classes. The general distribution of these features can be used to differentiate object classes. A very insightful treatment of this idea is provided in [5]. An archetypal bag-of-words approach involves extracting image features such as SIFT[20], SURF [2], color histograms etc., from a large number of training images. The features can be extracted by dense sampling or grid sampling,

or can even be extracted for the entire image. In the case of SIFT and SURF, the inherent feature detector which detects keypoints gives rise to the descriptors. Once the features are extracted they are clustered to a lower number of feature centers using a clustering algorithm like K-means or dictionary learning. Then the descriptor for each training image is recomputed based on the statistics of the clusters in the codebook that represent the features in the image. Discrimination is then achieved by using a classifier such as K nearest neighbors or support vector machines (SVM). For support vector machines, we use the radial basis function kernel. We determine the parameters for the kernel and the codebook size via cross-validation on the training set images. The features used in our approach are SIFT, SURF for their speed and robustness, and PHOG which can be used as a descriptor for the entire image and can be considered a good descriptor for object classification [6].

The way the vocabulary is constructed is different for different kinds of features. For the SURF and SIFT features, the codebook is obtained by clustering a large number of features into a set of codebook vectors. For the PHOG method the descriptive power of the vocabulary is directly affected by the number of levels in the pyramid, as well as the number of orientation bins in the histogram. The single PHOG descriptor for the entire image (blob) is computed and stored directly in the vocabulary. We already acquire a histogram for each image; hence no further clustering is required. The angle range parameter for PHOG computation was fixed to 360 degrees. The other parameters such as number of pyramid levels and number of orientation bins are all determined through

cross-validation on the training set. An illustration of the processing pipeline is shown in Figure 6.2.

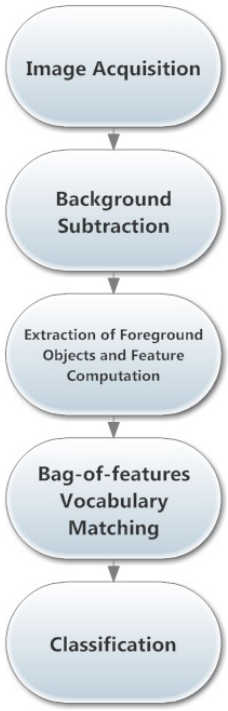

Figure 6.2: Flowchart depicting the processing steps on each image frame from a video sequence.

6.3.2 Bag of Salient Words

The bag of salient words is a modification of the bag-of-words approach for classification. Instead of using features obtained from the images directly, we sample only the top $M\%$ salient regions for descriptor computation. In order to measure the saliency of each region in the image with respect to the entire image we use the self-similarity score

of each region. The self-similarity of different regions of an object is considered to be informative in determining the class of object. Shechtman and Irani [42] and Deselaers and Ferrari [43] introduced the ideas of local and global self-similarity respectively. We developed a novel self-similarity measure [8] based on the sparse representation error of each region in the image with respect to the entire image. The larger the error the greater is the dissimilarity of the region with respect to the entire image. These large error regions can be thought of as the most salient regions (see Chapter 3), as they are least like the rest of the image. We find the top $M\%$ salient regions by taking the regions with the top $M\%$ highest error values. We then compute the region covariance [7] and pyramidal histogram of oriented gradients [6] descriptors. With these descriptors, we proceed to generate bag-of-words descriptors for the images of bicyclists and pedestrians as discussed previously in Section 6.3.1 using the obtained salient descriptors. In [8], we propose two descriptors which model the distribution of saliencies and can be directly used for classification. However for real traffic images, we resorted to more robust descriptors such as the pyramidal histogram of oriented gradients and region covariance descriptors. This is because the self-similarity descriptors are not robust to occlusion and view changes. The SIFT and SURF feature detectors can be thought of as local saliency detectors whereas our approach detects global saliency. Since this information can also prove to be useful in the discrimination of object classes, we incorporate this approach in our appearance model.

Samples of the saliency maps obtained for bicyclists and pedestrian images are shown

in Figure 6.3.

6.3.3 Combined Approach

We have detailed our individual approaches thus far. The blob image can be analyzed rigorously using the bag of visual words method that employs interest point features (SIFT, SURF, etc.) and holistic descriptors such as PHOG. Further, we add local image analysis performed by discriminative dictionary learning and sparse coding [56]. Each classifier estimates a probability of whether the blob is a bicyclist or a pedestrian. Then the individual predictions can be combined using a Naive Bayes classifier as in Equation (6.1).

$$\text{class label} = \arg\max_{\text{class}} \left(\prod_{i=1}^{n} P\left(X_i|\text{class}\right) \right) P\left(\text{class}\right), \tag{6.1}$$

where $P(\text{class})$ denotes the class prior (uniform), and $P\left(X_i|\text{class}\right)$, $i = 1, \ldots, n$ denotes the n individual classifier likelihoods. When combining using the Naive bayes approach the individual dimensions are assumed to be independent. Each probability is a result of the corresponding SVM classifier output or the discriminative dictionary classifier output. Note that the classifier output from an SVM is converted to probability estimates using regression models. More information can be found in [64]. We used the Libsvm library for our implementations [65].

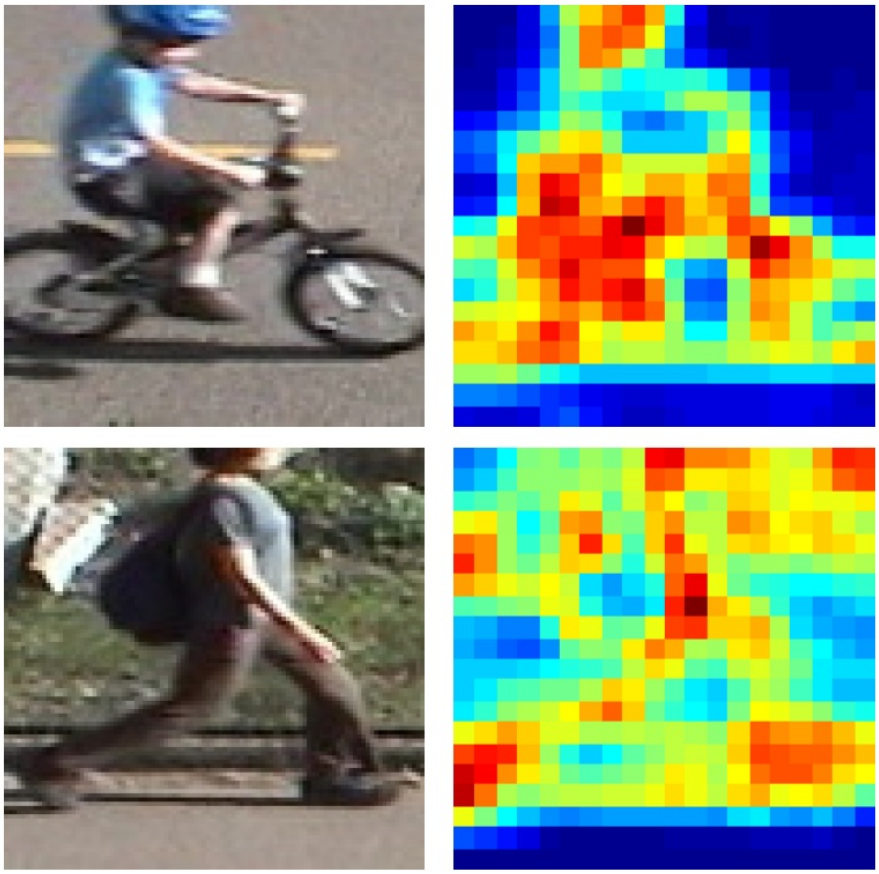

Figure 6.3: Saliency maps showing the saliency patterns found in the images of a bicyclist and a pedestrian. Red regions indicate high saliency (low self-similarity) and blue regions indicate low saliency. Yellow regions indicate intermediate regions.

Alternatively, inspired by the work in [66], we also combined the best individual bag-of-features methods by concatenating the histogram feature originating from each feature descriptor into one histogram. Then we re-normalized the concatenated histogram to form a new global histogram descriptor for the entire image. These descriptors were then used in a SVM classifier. We employ this approach because not all dimensions of the individual feature descriptors are always informative. By concatenating the different histogram descriptors into one descriptor the classifier interprets the most useful feature dimensions from all possible descriptor dimensions. Also for the Naive Bayes approach, we combine the three best bag-of-features SVM classifier outputs (SURF, SIFT and PHOG) along with the discriminative dictionary learning classifier. However, for the concatenated histogram approach we only use the three best bag-of-feature histograms and this then used in one SVM classifier. We provide a comparative analysis of all the methods in Section 7.3.

6.4 Software Implementation and Time Complexity

An overview of the practical implementation of a classification and counting system is shown in Figure 6.2. The implementation was carried out in C++ using open source computer vision libraries such as OpenCV and VXL. This can be employed with live camera streams or with recorded videos. The first step of processing is background removal and separation of foreground objects. This is done using the mixture of Gaussians method of background modeling [67]. The separated blobs are associated across

different frames using a bipartite graph based approach which is based on blob area overlap percentages. Although tracking blobs is not of primary interest to this problem, it provides useful information when combined with the calibration information. The scene was calibrated beforehand using the method described in [68].

The time complexity of the K-SVD dictionary learning algorithm is analyzed extensively in Rubinstein *et al.* [69], and is mentioned here for the reader's convenience. The operation count of the K-SVD algorithm is given by:

$$T_{\mathrm{K-SVD}} = R \times (2NL + K^2L + 7KL + K^3 + 4KN) + 5NL^2 \tag{6.2}$$

per iteration, where R is the number of training signals, K is the target sparsity, N is the dimensions of the signal, and L is the number of atoms in the dictionary. In the usual case where $K \ll L$ and $N \ll R$, we can approximate this as

$$T_{\mathrm{K-SVD}} \approx R \times \left(K^2L + 2NL\right). \tag{6.3}$$

The dictionaries are however computed offline in batch mode. The classification step involves determining the reconstruction error path, which can be determined efficiently using the OMP method, which has a per-signal time complexity of [69]

$$T_{\mathrm{OMP}} = 2NL + K^2L + 3KL + K^3. \tag{6.4}$$

With regard to computing the bag-of-words features, since the blobs are relatively small the time taken to compute the SIFT, SURF, and PHOG features is considerably low. Even though all our experiments are performed in Matlab for analysis, a full fledged

implementation in OpenCV can be carried out. Upon testing a C++ implementation on Intel core i5 machine running Windows 7 64 bit, we obtained the processing times shown in Table 6.1. The times shown are those for processing 3 blobs of size $\approx 150 \times 150$ pixels. A total of 627 milliseconds can be expected for a typical frame of size 720×480 pixels. However, during our analysis, we only do the processing once every 10 frames. This implies the average processing time is $\frac{1 \times 627 + 9 \times 120}{10} = 171.7$ milliseconds as we incur the full processing time only once every 10 frames. For the remaining 9 frames, we only incur the foreground extraction and tracking cost of 120 ms. This results in a processing speed of ≈ 6 frames per second. The processing speed can be further increased by optimization and reducing the frame size / rate. Note that the salient features are slightly expensive to compute; however they enhance the performance of the method. Optionally, they can be left out to improve the speed of the method.

6.5 Experiments

In this section we discuss the experiments we conducted with details about the data, parameter selection and a comparative analysis with another approach based on the work presented in [70].

6.5.1 Parameter Selection

The descriptive power of the codebook of SIFT and SURF features is varied by changing the codebook size. The effect of varying codebook size on the classification accuracy is

Step	Time in milliseconds
Foreground extraction & Tracking	120
SIFT computation	97
SURF computation	10
PHOG computation	35
Vocabulary matching	40
SVM prediction	25
Sparse coding	10
Self-Similarity / Saliency	300
Total	**627**

Table 6.1: Average time of processing for one frame assuming the presence of 3 blobs of size $\approx 150 \times 150$ pixels.

Codebook Size	Accuracy
500	96.5%
1000	97.86%
2000	98.26%
4000	**99.06**%
5000	98.46%
6000	97.2%

Table 6.2: Performance of SVM-SURF as function of the parameters on the training set.

indicated in Tables 6.2 and 6.3. A tabulation of classification accuracy as a function of these parameters is presented in Table 6.4. For this analysis we acquired a training set from real world data (bicycle trail video). We processed about an hour of video to remove the background using the mixtures of Gaussians approach and then performed tracking on the extracted foreground blobs. Of the extracted blobs, we use about 700 images (of unique blobs) for training with a near equal split for bicyclists and pedestrians. These training images were consistently used for determining all the parameters of the different approaches presented in this chapter. Some samples from the training set and their corresponding foreground masks obtained through foreground detection are shown in Figure 6.4.

Codebook Size	Accuracy
500	93.85 %
1000	96.39 %
2000	**97.72**%
3000	92.43 %

Table 6.3: Performance of SVM-SIFT as function of the parameters on the training set.

Pyramid	Number of bins		
Levels	6	8	10
1	88.1 %	**93.4**%	87 %
2	87.2 %	90.9 %	86.3 %
3	82.4 %	86.5%	80.1 %
4	79 %	81.1 %	76.5 %

Table 6.4: Performance of SVM-PHOG as function of the parameters on the training set.

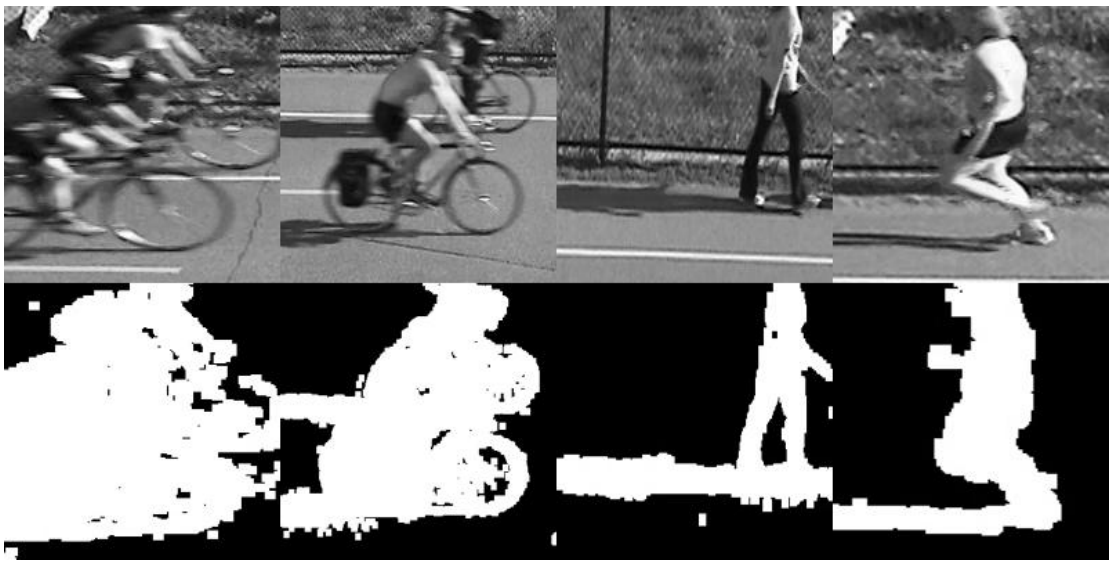

Figure 6.4: Samples from the training set and the corresponding foreground mask.

Based on the cross-validation analysis on the parameters for the bag-of-words approach, we concluded that for the SURF and SIFT features, the codebook size was chosen to be 4000 and 2000 respectively. For the PHOG feature computation, we chose a 1 level pyramid, with 8 orientation bins. The accuracies reported here are those of the corresponding best performing SVMs with a tuned RBF kernel (chosen from 3 possible choices of kernels: linear, polynomial and RBF). The kernel parameters were chosen through grid-search and cross-validation. For example in the case of the SURF bag-of-features, an RBF-kernel SVM with $C = 256$ (error penalty) and $\gamma = 0.0625$ (coefficient in the exponent of the RBF kernel) was used.

Patch Size	% Salient Considered		
	10	20	30
10	88.1 %	89.2 %	88.4%
12	87.1 %	87.4%	87 %
14	85.9%	84.8%	84.8 %

Table 6.5: Performance (classification accuracy) of salient bag-of-words with pyramidal HOG descriptors as a function of patch size and % saliency considered.

6.5.2 Salient Bag-of-Features

Similarly, for the computation of the global saliency using self-similarity the parameters were estimated using cross-validation on the training set. Overlapping patches of sizes 10, 12, and 14 were considered, and the best performance was obtained using a patch size of 10 and using covariance descriptors. Classification accuracies shown in Tables 6.5 and 6.6 are the best average 10-fold cross validation accuracies on the training set with the best choice of SVM parameters. We note that the performance does not change by a lot while changing the patch sizes. This is inherent to the multi-scale manner in which the saliency is measured. Hence for testing as well as in the combined classifier, salient covariance descriptors for patch size of 10×10 of the top 20% salient regions were used.

Patch Size	% Salient Considered		
	10	20	30
10	92.1%	92.7%	92.01%
12	90.3%	91.1%	90.8%
14	87.3%	87.3%	86.6%

Table 6.6: Performance (classification accuracy) of salient bag-of-words with region covariance descriptors as a function of patch size and % saliency considered.

6.5.3 Classification and Counting Results

For testing, we used two full video sequences: the first sequence is from a bicycle trail (Gateway Trail) in Minneapolis [1] and the other is from a university walkway. Sample images from these sequences are shown in Figure 6.5. Since the first sequence is from a bicycle trail, there was a much higher traffic of bicyclists than pedestrians (338 bicyclists and 53 pedestrians). On the contrary, the university sequence had a much higher pedestrian presence (268 pedestrians and 52 bicyclists). The individual track sets (sequences of blobs) are annotated manually as a bicyclist or a pedestrian. When the testing is performed, the classification decision for the entire track set is made based on the classification decision of the individual frames using the most frequent classification occurrence (mode).

[1] `http://www.dnr.state.mn.us/state_trails/gateway/index.html`

The test sequences present two extreme counting scenarios and present a lot of diversity in the data. The bicycles in Gateway Trail move much faster than in the university scenario. Also pedestrians in a bicycle trail are often jogging, roller-blading, or sprinting whereas in the university they are walking at a slow pace. In trails, pedestrians are likely to be individuals whereas in the university scenario, a lot of crowds and groups can be observed. These issues make simple cues of the objects like size, velocity, perimeter etc. unreliable to be used across multiple scenarios. Intuitively, the appearance alone is stable and repetitive across different challenging situations. When describing appearance, issues like occlusion, scale changes, are generally critical. Hence we resort to strong scale and affine invariant features in SURF and SIFT. The PHOG features are useful in providing an overall description of the blob. The bag-of-words model is also helpful in learning models invariant to these changes. The discriminative dictionary learning is also built around a multi-scale framework.

However, we notice in Figure 6.5 that the view changes are not so significant in our tests (fixed single camera with limited field of view). We speculated that simple features might actually perform adequately well in this data. However, the results indicate otherwise. We have shown the results of our approach and compared it with an approach presented in [70]. This approach uses blob morphological properties to filter candidate blobs as pedestrians and bicyclists. We made some minor modifications to the approach. Instead of deriving thresholds for the parameters, we used a support vector machine to classify the blobs as pedestrians or bicyclists based on the morphological properties

such as area of the blob, convex area of the blob, eccentricity, solidity of the blob and perimeter. These properties define the morphological shape and size description of the objects. These features can be surprisingly powerful; however they can suffer severely from occlusion and tracking artifacts like blob merging and splitting. In order to train the classifier, we used a slightly different scheme. Since we did not possess calibration of the scene, the same classifier cannot be used in multiple scenarios. The size and velocity of blobs in image pixel units are functions of the camera's intrinsic and extrinsic parameters. Hence we trained separate classifiers for the different videos. For the blob morphological properties we used 200 samples each of bicycles and pedestrians totaling 400 training images for each video. This gives the classifier about the same amount of information provided to the appearance-based classifiers. For the velocity training, we used about 50% of all data as training. A Gaussian distribution is approximated for the bicyclists and pedestrian velocities individually. The velocity distributions of bicyclists and pedestrains for the two video sequences are shown in Figures 6.6 and 6.7. Using these gaussians for each class—bicyclists and pedestrians—an estimate of the probability of the blob belonging to a class can be evaluated. Figures 6.8 and 6.9 show the area and perimeter distributions for the different datasets. Using Gaussians for each class (bicyclists and pedestrians), an estimate of the probability of the blob that belongs to a class can be evaluated.

The classification and counting results of the individual approaches and the Naive Bayes combined approach as well as combined histogram approach on the bicycle trail

Feature Classifier	Frame Classification Accuracy	Counting Accuracy
SURF + BoW + SVM	86.26%	92.07%
SIFT + BoW + SVM	80.3%	91.3%
PHOG + SVM	82.54%	93.35%
Discriminative Dictionaries	84.44%	89.26%
Combined Naive Bayes	86.62%	94.88%
Blob Morphology + SVM	78.13%	89.76%
Velocity Distribution	-	64%
Morphology + Velocity	-	66.24%
Covariance K-means (K=10) [71]	89.57%	93.96%
Covariance K-means (K=5) [71]	88.68%	91.53%
Salient Bag-of-Covariance	83.8 %	88.7%
Combined Naive Bayes + Salient Bag-of-Covariance	88.1 %	95%
Combined histogram + SVM	**90.3**%	**95.4**%

Table 6.7: Performance on bicycle trail data.

Feature Classifier	Frame Classification Accuracy	Counting Accuracy
SURF + BoW + SVM	88.85%	93.12%
SIFT + BoW + SVM	80.94%	86.87%
PHOG + SVM	53.8%	55%
Discriminative Dictionaries	65.61%	75.31%
Combined Naive Bayes	92.03%	93.75%
Blob Morphology + SVM	17.59%	9.37%
Velocity Distribution	-	94.69%
Morphology + Velocity	-	94.69%
Covariance K-means (K=10) [71]	91.03%	85.5%
Covariance K-means (K=5) [71]	88.54%	84.25%
Salient Bag-of-Covariance	84.63 %	89.21%
Combined Naive Bayes + Salient Bag-of-Covariance	93.01 %	94.31%
Combined histogram + SVM	**93.81**%	**95.07**%

Table 6.8: Performance on university walkway data.

Feature Classifier	Frame Classification Accuracy	Counting Accuracy
SURF + BoW + SVM	87.56%	92.6%
SIFT + BoW + SVM	80.62%	89.09%
PHOG + SVM	68.17%	74.18%
Discriminative Dictionaries	75.03%	82.28%
Combined Naive Bayes	89.32%	94.31%
Blob Morphology + SVM	47.86%	49.56%
Velocity Distribution	-	79.34%
Morphology + Velocity	-	80.46%
Covariance K-means(K=10) [71]	90.3%	89.73%
Covariance K-means(K=5) [71]	88.61%	87.89%
Salient Bag-of-Covariance	84.21 %	88.95%
Combined Naive Bayes + Salient Bag-of-Covariance	90.55 %	94.65%
Combined histogram + SVM	**92.05**%	**95.23**%

Table 6.9: Overall results.

Figure 6.5: Top: Images from the university walkway sequence. Pedestrians crowds are more common making classification through morphological properties difficult. Bottom: Images from the bicycle trail. Pedestrians and bicyclists are isolated making it easier to classify with morphological properties. However, the velocities are more confusing due to activities like roller-blading,sprinting and jogging.

and the university walkway are shown in Tables 6.7 and 6.8 respectively. The overall results which combines both datasets are shown in Table 6.9. The combined Naive Bayes method combines the best individual SURF, SIFT BoW and PHOG SVM classifiers determined from training set performance. The combined histogram method combines the best individual SIFT, SURF BoW, and PHOG histogram descriptors, along with the salient covariance bag-of-words histogram descriptor also determined based on training

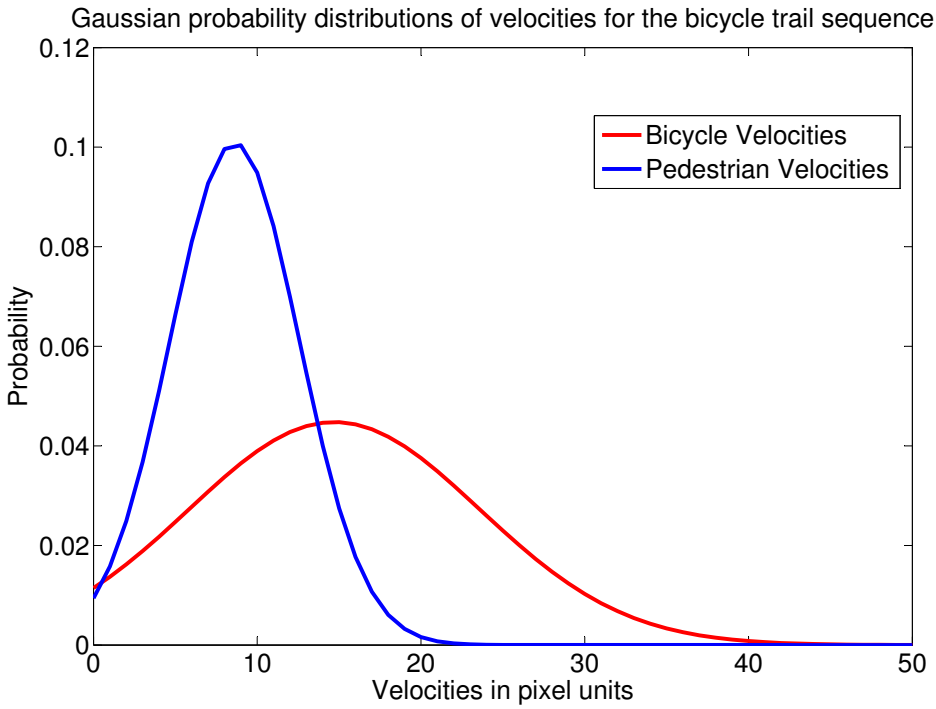

Figure 6.6: Velocity distribution of bicyclists and pedestrians in the bicycle trail. See Table 6.7 for velocity based classification accuracy.

set performance. The frame accuracy reported are the classification accuracies of each individual frame considered for each blob. There were a total of 2058 frames classified in the bicycle trail sequence and 6505 frames classified in the University walkway sequence. We notice how the appearance based methods are consistent in their performance compared to the classification based on blob morphological properties and velocity. The velocities proved to be useful in the university walkway because pedestrians moved at a distinctly slower pace than bicyclists. However, morphological properties [70] did not

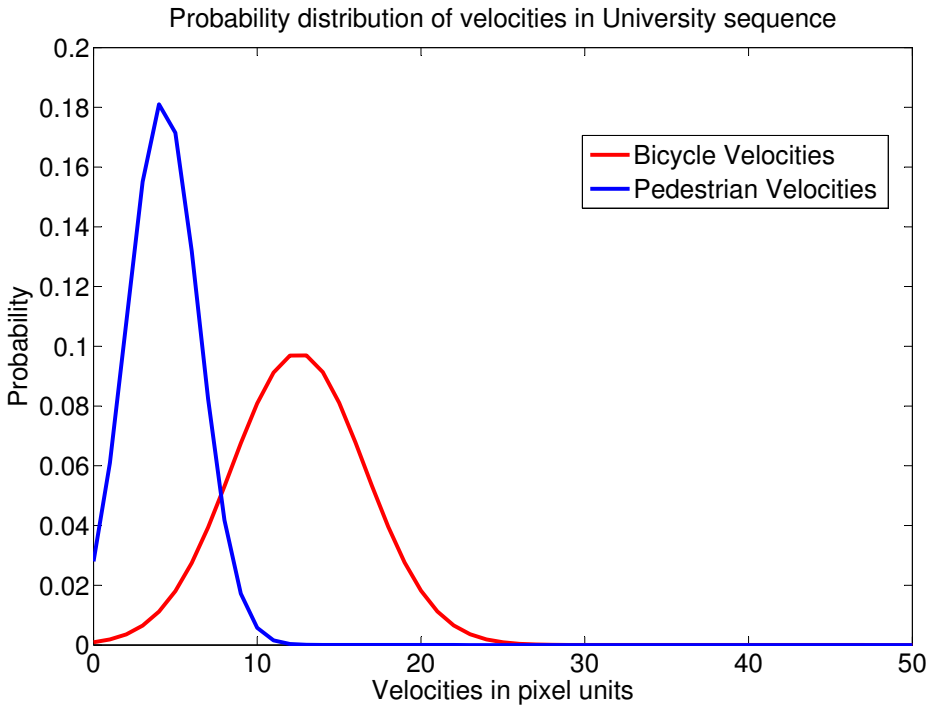

Figure 6.7: Velocity distribution of bicyclists and pedestrians in the University walkway. See Table 6.8 for velocity based classification accuracy.

prove useful owing to phenomena such as walking in groups, crowds, walking bicycles etc. However, in the bicycle trail the velocity was less useful because of the different behavior of pedestrians. The morphological properties were more useful because the bicyclists and pedestrians were isolated from each other most of the time, hence a clean blob could be almost always extracted. Only the appearance based method consistently performs across both sequences. And in both occasions the combined Naive Bayes approach and the combined histogram approach were able to provide the best results in

both frame classification and counting. The combined approach is useful because for each frame if one classifier is not confident based on the probabilities, and the rest of the classifiers are more confident, then the majority dominates the final classification result. This way for each frame we will make a mistake only if most or all of the classifiers are wrong. Since the combined classifier is better than any of the individual classifiers, it shows that all the individual classifiers contributed to the information obtained, thereby establishing the efficacy of both global and local image analysis. We also note that the inclusion of the salient covariance descriptors bag-of-words method improves the performance even more. This is because this method captures globally salient features which are not captured by either the SIFT or SURF features. The classification accuracies presented for the method in [71] are the results of a modified approach. The authors originally perform classification via cascading logitboost classifiers on the covariance descriptors from different parts of a candidate region. Whereas this has proven to be highly useful in detection to reject false alarms and improve hit rate, for our problem since we have the candidate region of interest we only need to perform the geodesic distance-based matching. Hence we performed K means clustering using the geodesic distance presented in their paper that is valid for Riemannian manifolds of positive definite matrices (covariances). We report the average accuracy of 10 iterations of K-means with the value of K chosen to be 5 or 10. We used the same training and test data as other methods. We notice that this approach obtains very high frame classification accuracy; however it is more sensitive to issues with tracking. This is reflected in the

lower counting accuracy.

6.6 Conclusions

In this chapter we proposed a multi-modal multi-scale appearance based blob classification approach. We also presented a Naive Bayes framework as well as a combined histogram approach with which the different approaches can be combined to improve upon the performance of each of the individual methods. With the strong classification results and with the aid of tracking we could address the problem of classifying and counting composite objects such as bicyclists amongst simpler objects like pedestrians. We also compared our approach with a blob morphology and velocity classifier, based on [70]. We achieved better results overall with an average frame-by-frame classification accuracy of $\approx 92\%$, and a counting accuracy of $\approx 95\%$. Our results indicate the importance of appearance based methods and the significance of both global and local image analysis. Our experiments were carried out in two different scenarios which span conditions in which pedestrians and bicyclists can present themselves in a completely different way in terms of morphological structure and velocity. However, appearance is the only stable feature across all scenarios. Our evaluations were also rigorous as they were based on classifying 321 unique pedestrians and 390 unique bicyclists covering a total of approximately 9000 individual frames. Future work involves improving counting accuracy in crowds and groups based on the work in [72].

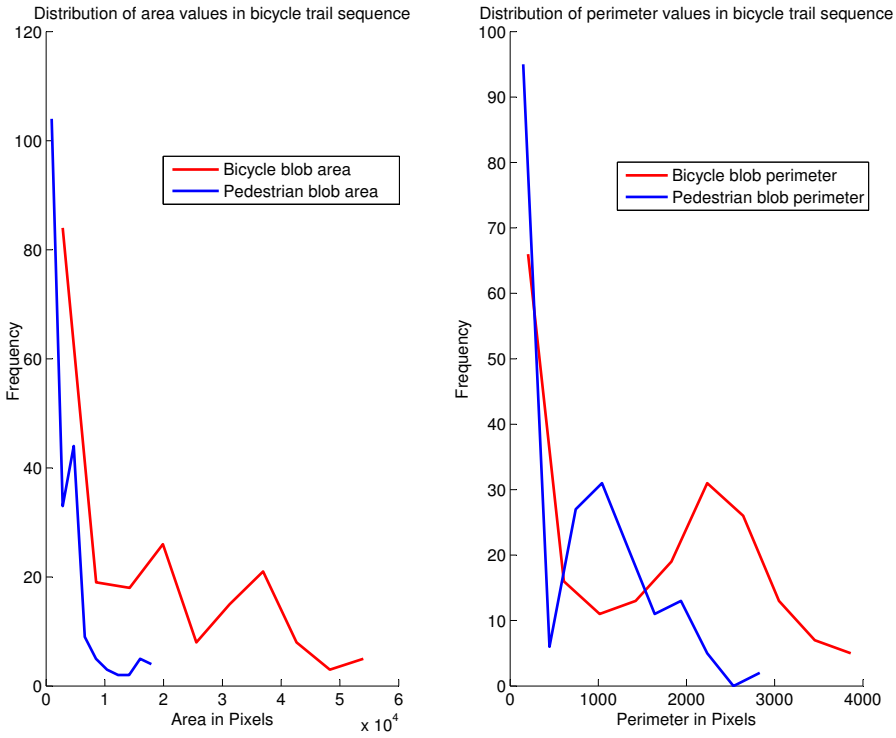

Figure 6.8: Area and perimeter distribution of bicyclists and pedestrians in the bicycle trail. Area values are separated well however the perimeter values are not. See Table 6.7

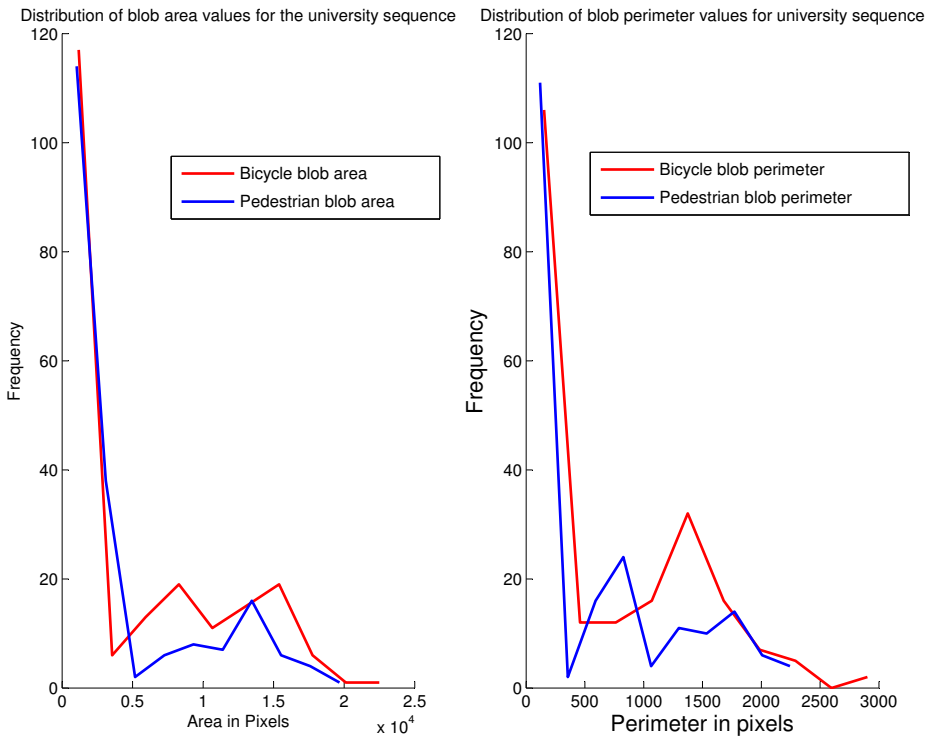

Figure 6.9: Area and perimeter distribution of bicyclists and pedestrians in the University walkway. Both area and perimeter distributions are not separated well. See Table 6.8.

Chapter 7

Action Recognition Using Global Spatio-Temporal Features Derived from Sparse Representations

The performance of classification methods has been strongly influenced by the choice of feature detectors, descriptors, as well as the classification algorithm. For the purpose of classification of actions in video sequences, local spatio-temporal feature detectors along with descriptors such as HOG (histogram of oriented gradients), HOG3D (spatio-temporal HOG), HOF (histogram of optical flow) have been used with success before.

In contrast we propose the use of global spatio-temporal features detected based on a spatio-temporal self-similarity measure using sparse representations. The purpose of detecting global spatio-temporal features is to reduce the redundancy in the modeling stage and to focus attention on the most informative regions. Only the top $M\%$ salient (least self-similar) regions are considered. This is followed by the computation of the histogram of oriented gradients and region covariance descriptors of the spatio-temporal salient regions. The ensemble of such block descriptors in a bag-of-features approach provides a holistic description of the motion sequence which can be used in a classification framework like SVM. This approach can be thought of as dense sampling followed by salient region detection. The proposed feature detector also assumes a formulation which is practically feasible and computationally efficient. Our approach performs competitively with the state-of-the-art on the KTH action dataset, the UCF Sports Action dataset, as well as the Hollywood Human Actions dataset for human action classification.

7.1 Introduction

Action classification in video sequences has received a lot of attention from the computer vision research community as a problem independent of object recognition in static images. Even though conceptually the problems are similar, object recognition techniques in static images have not been readily generalized to solve video classification problems. While issues such as intra-class variations, scale, affine transformation, noise, clutter,

etc., that are challenges for object classification also pose problems in video classification, we also encounter additional constraints due to dynamic backgrounds, different rates at which actions are performed, and more expensive computational and storage demands. Explicit data-driven approaches for video classification methods are still not very common due to prohibitive computational needs. However, the "data-deluge" has manifested itself even in the realm of videos due to popular video hosting websites such as Youtube, Dailymotion, etc. As in documents and images, the need for classification and clustering methods for video sequences has also risen rapidly.

Although there has been plenty of research in this area, not many approaches take advantage of the complexity or lack thereof (redundancy) in the video data. Laptev [73] introduced the space time interest point detector approach to finding the most unique spatio-temporal features in a manner similar to finding SIFT (scale invariant feature transform) features in images. This paper models significant local phenomena in the video based on spatio-temporal scale-space extrema. Alternatively one could think of the "interesting" regions as globally significant, i.e., given the spatio-temporal variations of the entire video sequence, the top $M\%$ most informative variations could be of significance. The idea is that even if there is motion in the background due to a moving camera scenario (for example), anything of significant foreground motion will induce different spatio-temporal variations than the rest of the video. Capturing such variations are very useful for cases such as the Hollywood movie actions dataset [74] where the actions of interest are often performed amidst significant background

variation.

In order to determine which spatio-temporal variations are informative, we take the approach of modeling the complexity of each spatio-temporal patch (a sub-window of the image sequence). Theoretically speaking, we would like to estimate the Kolmogorov complexity of each spatio-temporal patch. Even though the Kolmogorov complexity can not be computed in practice, we can attempt to approximately estimate it by way of measuring each patch's representation length with respect to a basis under some sparsity assumptions. In fact, we can observe in practice that many signals are redundant, and it is fair to assume that such sparsity exists if a suitable basis is chosen. Most patches can potentially be represented sparsely with a suitable basis; however, despite efforts, there might be a few patches which cannot be represented sparsely. Such patches might require additional representation "budget", or in other words their representation lengths are longer than expected. These patches are likely to be more informative (less redundant). The exact values of the estimates are not important, but only the ordering of the saliences, since our interest is to rank the top $M\%$ salient patches.

This chapter is organized as follows. In the next section significant prior approaches in the literature for the problem of human action classification based on spatio-temporal features are discussed. We do not limit our discussion only to methods based on Space-Time Interest Points/Regions approach but also other generic data driven methods. In Chapter 3, we provide the theoretical motivation for measuring the complexity (description length) of spatio-temporal patches. We formulate the problem of measuring the

complexity as the representation error (ℓ_2 norm distortion) based on dictionary learning and sparse representation. By using these methods, we can take advantage of the already existing optimized mathematical tools (KSVD [47], OMP [32], etc.) by using this approach. In Section 7.2.2 we provide an algorithmic description of our approach for classifying human actions using the proposed global spatio-temporal features in a bag-of-features framework. Finally in Section 7.3, we discuss the performance of our approach comparatively on the KTH human actions dataset, the UCF sports action dataset, as well as the Hollywood movie actions dataset.

7.2 Prior Work

Action classification has a wide variety of applications such as surveillance, traffic monitoring, Internet video summarization, automatic captioning, etc. Hence many researchers approach this problem from different viewpoints motivated by their primary application. However, with the course of time there has been a steady exchange of ideas and several permutations of different representation and classification schemes have been made possible. This chapter describes a representation scheme akin to space-time interest point models. There has been a lot of work in the past advocating space-time interest point models. Interestingly almost all of these methods [73, 75, 76] have employed a local spatio-temporal scale-space extrema detection approach. In contrast we argue that significant spatio-temporal features can be obtained by determining the most informative spatio-temporal regions of a video sequence. Even though, at the outset

global feature detection might appear expensive, with the aid of appropriate tools these features can be estimated very efficiently.

There are other popular methods of representation, such as the class of dynamic models captured by hidden Markov models (HMM). HMM have been successful primarily due to the fact that they model the nature of variations over time inherent to different types of actions. In [77], a rigorous survey of HMMs with different kinds of state space representations is provided. Bashir *et al.* [78] demonstrated the use of HMMs with spatio-temporal curvature representations and PCA decompositions for trajectory based human action classification. Another powerful representation method involves describing actions as a sequence of shapes, or in the case of human actions, silhouettes [79]. This method has gained a lot of popularity due to its invariance properties.

Guo *et al.* have made different contributions based on sparse representation in covariance manifolds of optical flow features [80] as well as silhouette tunnels [81]. Castrodad *et al.* [82] proposed a deep-layered discriminative approach to classifying human actions based on learned basis vectors or action primitives for each action category. These approaches try to capture the most non-redundant or informative spatio-temporal features representative of a certain class (top-down). In contrast, we determine in a bottom-up manner the global spatio-temporal features corresponding to the most non-redundant (least self-similar) regions of a video sequence. The discriminative modeling in our classification step is performed using a support vector machine. There are many other forms of interest region detectors for video sequences which are usually an extension

of 2D interest point detection methods to spatio-temporal sequences. Some examples are Harris 3D [73], Cuboids [75], Hessian [83] and dense sampling [84, 85, 86]. Detailed evaluation surveys of local spatio-temporal feature detectors as well as descriptors are provided in [87, 88].

7.2.1 Formulation

In this section we provide a brief introduction to the theory of dictionary learning and sparse coding. In this framework we attempt to represent a signal $x \in \mathcal{R}^n$ using a suitable basis/ dictionary D with a small number of coefficients α_i corresponding to the columns d_i of D. Such a dictionary can be learned by solving the following optimization problem:

$$\min_{\alpha,\mathbf{D}} \sum_{i=1}^{M} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 \;\text{ s.t. }\; \|\alpha\|_0 \leq L. \tag{7.1}$$

Given a dictionary D, the sparse representation error of a signal x is given by

$$\mathcal{R}(\mathbf{x}, \mathbf{D}, \alpha) = \|\mathbf{x} - \mathbf{D}\alpha(\mathbf{x}, \mathbf{D})\|_2. \tag{7.2}$$

Here $\alpha(\mathbf{x}, \mathbf{D})$ is the L-sparse decomposition for the pair $(\mathbf{x}, \mathbf{D})$. Learning the dictionary D as well as recovering the sparse coefficients α_i for a signal x_i are both non-convex problems. However, efficient convex approximate methods such as KSVD [47] can be used for the learning procedure. The sparse recovery can be done using a greedy algorithm known as the orthogonal matching pursuit [32], for which we can leverage the

already existing efficient implementations [49, 50]. We resort to these approaches for reasons related to efficiency rather than correctness or exact recovery. The reason is that we are interested in quickly determining the sparse representation error (residue) rather than the best sparse representation (coefficients), since we are dealing with a large amount of 3-dimensional data (spatio-temporal volumes).

7.2.2 Approach

We propose the use of sparse representations on spatio-temporal patches to determine the saliency of those patches. We argue that these globally salient spatio-temporal regions are very informative while modeling action sequences. That is the saliency of each spatio-temporal volume is measured with respect to the rest of the video sequence itself rather than just the local neighborhood. This is unlike many contemporary approaches for detecting spatio-temporal features such as the Harris 3D [89] and STIP (spatio-temporal interest points) [73]. However, we share a similarity with these approaches in our effort to make the features as robust to spatio-temporal scale variations as possible. We perform convolution of the spatio-temporal patches with a spatio-temporal (3-dimensional) Gaussian kernel at two different scales to form a 3-level Gaussian pyramid (including the native scale). Adding more scales can improve the robustness of the approach; however it increases the processing time of each spatio-temporal window.

Feature Detection (Salient Region Detection)

Each video sequence is first arranged into many sliding temporal windows with w number of frames in each window. This window is then arranged in a Gaussian pyramid as mentioned before. Each pyramid level is then densely sampled into spatio temporal patches of size $b \times b \times w$ as shown in Figure 7.1. The spatio-temporal patches are then vectorized to form the signal matrix X and each signal is a column and is b^2w dimensional. Three such signal matrices are produced corresponding to the three-level Gaussian pyramid. Then a dictionary (basis) is learned separately for each scale. These dictionaries are then concatenated column-wise into a single multiscale dictionary. Then we proceed to measure the sparse representation error by performing orthogonal matching pursuit to determine the best k atoms from the dictionary to represent each spatio-temporal patch in the native scale. The best k atoms that are chosen could thus be originating from different scales. The residual error in such a representation is robust to variations in scale within the assumed limits. The multi-scale dictionary spans the entire vector space of the spatio-temporal window under the sparsity consideration. In other words, the dictionary minimizes the average error in representation of every spatio-temporal block while using only k of its columns to represent each block. This implies that not all spatio-temporal blocks will have low representation error. The relative residual error for each patch (ranging from 0 to 1) is used to then rank the top $M\%$ salient patches. The higher the residual error, the more self-dissimilar the patch is with respect to the entire spatio-temporal window and the more salient it is. After these top regions are

selected we compute the corresponding spatio-temporal descriptors.

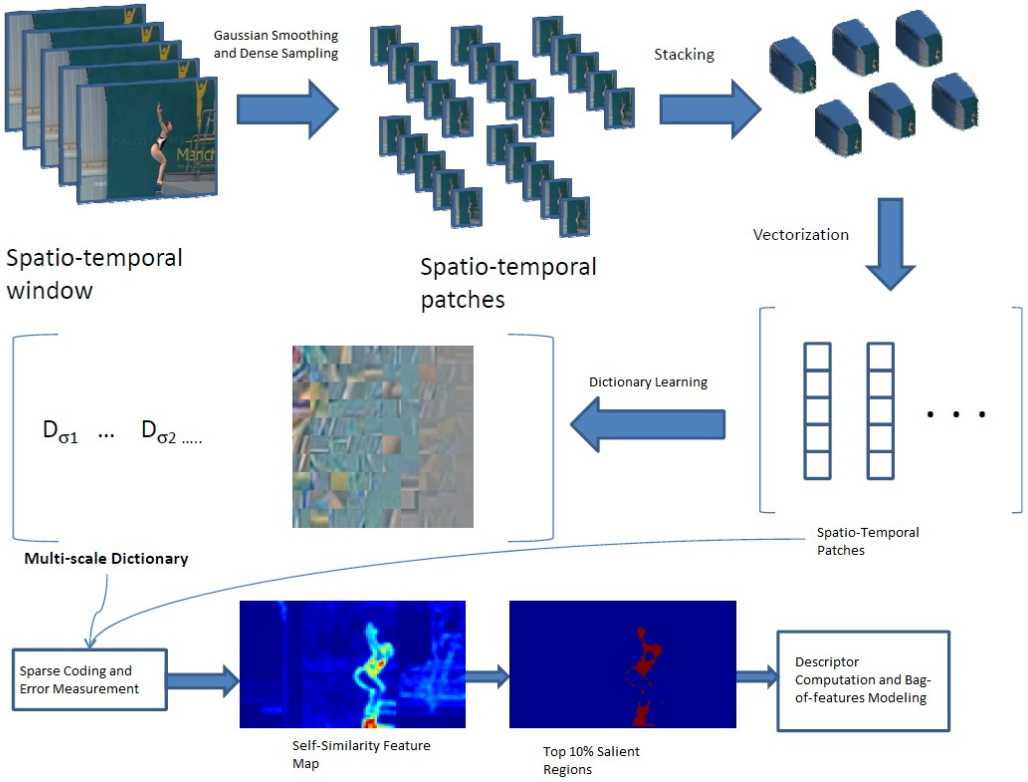

Figure 7.1: An illustration of the approach.

Descriptor Computation

After the salient regions are detected, the following descriptors are computed: HOG (histogram of oriented gradients [21]) and the region covariance descriptor [7] (extended for spatio-temporal volumes). While computing the HOG feature descriptor of the spatio-temporal patches, the HOG feature of each $3 \times 3 \times 2$ sub-patch is computed individually and then concatenated with re-normalization to form the HOG descriptor

of the entire spatio-temporal patch.

The following features are used in the region covariance computation.

$$F(x,y) = [x,\ y,\ t,\ |\frac{\partial I(x,y)}{\partial x}|,\ |\frac{\partial I(x,y)}{\partial y}|,\ |\frac{\partial^2 I(x,y)}{\partial x^2}|, \\ |\frac{\partial^2 I(x,y)}{\partial y^2}|,\ |\frac{\partial^2 I(x,y)}{\partial x \partial y}|,\ |\frac{\partial I(x,y,t)}{\partial t}|]^T. \tag{7.3}$$

Here x, y, and t are the pixel locations in space and time coordinates. I is the gray scale intensity at (x,y,t), and the partial derivatives represent the first and second order gradients along the x, y, and time dimensions, respectively. We can also add more feature of interest such as optical flow to this set, providing more descriptive power. However, to minimize the computational burden, we do not wish to compute optical flow. Given this feature set at each x, y location, we can compute the covariance descriptor of a particular spatio-temporal patch as

$$C = \frac{1}{n-1}\sum_{i=1}^{n}(f_i - \mu)(f_i - \mu)^T. \tag{7.4}$$

We can then convert this covariance descriptor into a vector by first computing the matrix logarithm of C and then vectorizing the upper triangular part of the result [90]. This vectorized form can be compared using the Euclidean norm.

Bag-of-Features

The bag-of-features approach is commonly used for feature based object and action classification [87, 5]. For each dataset consisting of different action classes we detect the

top $M\%$ salient regions and compute the corresponding descriptors. We then collect $100k$ feature descriptors of each type (HOG, Region Covariance) and then we perform K-means clustering to generate a vocabulary. Using these vocabularies a histogram based on matching features for each action sequence is then computed. This histogram feature is then used as the feature descriptor for each action sequence in a support vector machine framework. For the SVM learning different kernels were tested: linear, Gaussian radial basis function (RBF), polynomial, and exponential χ^2 kernel:

$$K_{linear}(x, y) = x^T y + c \tag{7.5}$$

$$K_{polynomial}(x, y) = (\gamma x^T y + c)^d \tag{7.6}$$

where d is the degree of the polynomial.

$$K_{RBF}(x, y) = exp(-\gamma \|x - y\|_2^2) \tag{7.7}$$

$$K_{exp-\chi^2} = exp(-\frac{1}{A}\chi^2(x, y)) \tag{7.8}$$

where

$$\chi^2(x, y) = \frac{1}{2}\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{x_i + y_i} \tag{7.9}$$

defines the χ^2 distance between two histogram features. A is the mean χ^2 distance between all training samples [91].

In general, for the bag-of-features approach, the χ^2 or exponential χ^2 kernels are used since it is a better distance metric for histogram features. However, in some cases we noticed that the polynomial, or Gaussian radial basis function kernels, performed comparably. The best choice of the kernel was established through cross validation with each training set.

The SVM has a decision function of the form (for one class case):

$$f(x) = \sum_i w_i y_i K(x_i, x) - T. \tag{7.10}$$

Here i represents the index of the training sample. The kernel values involving the test and each training sample are combined using some learned weights w_i and then verified if above a threshold T. We evaluate datasets which are multi-class and we use multi-class SVMs [65] and report the performance with respect to each class.

7.3 Experiments and Results

In this section we provide a brief description of the datasets we used for evaluation. Then the process of selecting the best parameters for each dataset is discussed. Finally we provide details of the performance of our method on all the datasets.

7.3.1 Datasets

We use the KTH actions dataset [76], the UCF sports action dataset [92], and the Hollywood movie actions dataset [74].

KTH actions dataset: The KTH actions dataset has 6 action classes: Walking, Jogging, Running, Boxing, Waving, and Clapping. The dataset contains 2391 sequences of actions performed several times by 25 subjects in four different scenarios. The sequences have a frame rate of 25 frames-per-second and a resolution of 160×120 pixels. We used the training + validation set for training (16 subjects) and we report the performance on the test set (9 subjects). Some sample actions from this dataset and corresponding spatio-temporal saliency map (self-similarity), and the top 10% salient regions, are shown in Figure 7.2.

UCF sports actions dataset: The UCF sports action dataset consists of sequences collected from television channels such as the BBC and ESPN. The dataset contains 150 video sequences at a resolution of 720×480 pixels. The actions included are Diving, Golf swinging, Kicking, Lifting, Horseback riding, Running, Skating, Swinging, and Walking. These sequences were down sampled to a resolution to match the KTH actions dataset (160×120) in order to reduce the computational burden as well as to limit within which the spatial scale choices are allowed to vary. Also additional samples were created by using mirrored versions of the video sequences. Additionally, since no training or test set separation was provided, leave-one-out cross validation average performance is reported for this dataset. This is done by training the classifiers on all but one sequences and testing on the one sequence.

Some action sequences and corresponding salient features from this dataset are shown in Figure 7.3.

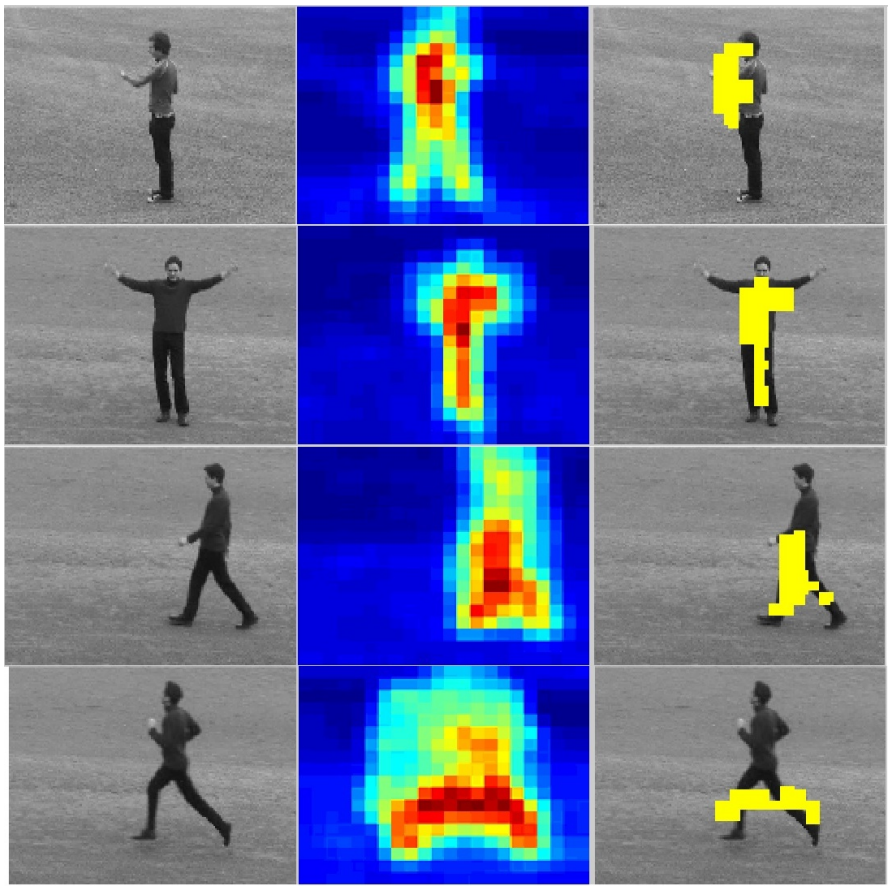

Figure 7.2: An illustration of some action sequences from the KTH actions dataset and the corresponding salient features. From left to right: an action frame, saliency map and the top 10% salient regions (used in feature computations).

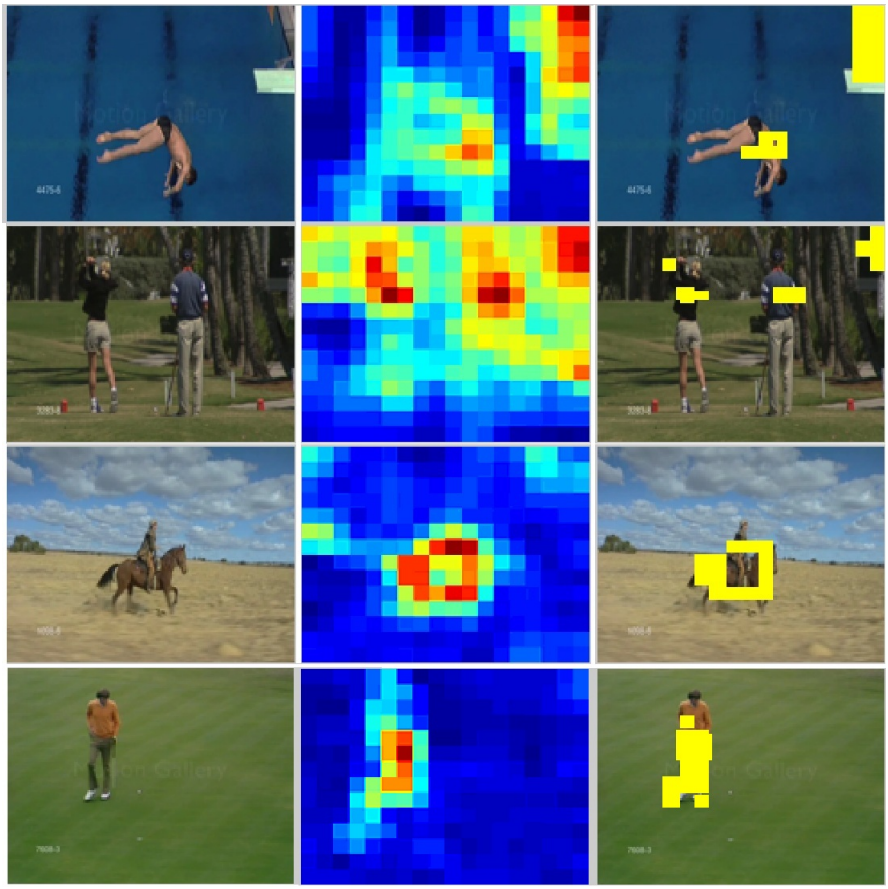

Figure 7.3: An illustration of some action sequences from the UCF sports actions dataset and the corresponding salient features. From left to right: an action frame, saliency map and the top 10% salient regions (used in feature computations).

Hollywood movie actions dataset:

The Hollywood movie actions dataset consists of more than 400 clippings from 32 movies representing the following actions: Answering Phone, Getting out of Car, Handshaking, Hugging, Kissing, Sitting down, Sitting up, and Standing up. We used the "train-clean" and "test-clean" sequences of this dataset for training our classifiers and testing. The training set consists of 219 action samples which contain manually provided labels obtained from 12 movies. The test set contains 211 labeled action samples from 20 movies. Some sample action sequences and corresponding salient regions are shown in Figure 7.4.

7.3.2 Parameter Selection

In order to obtain the best performance out of our approach, we need to determine the optimal choices for all parameters. By scaling all video datasets to the same size, we limit the ranges over which we need to search for the best choices. Specifically, the main parameters are: the spatial patch size (e.g., 14×14 to 20×20, the temporal window size (e.g., 4 frames in a window to 10 frames in a window), and lastly the choice of M for the selection of the top $M\%$ salient features. Even though we determine the features in a multi-scale approach, the same choices of parameters may not be optimal for all datasets. For example, the number of frames in a temporal window depends on how fast the actions are performed. In the KTH dataset, the actions are more structured and performed at a uniform rate. This cannot be expected in real sports

Figure 7.4: An illustration of some action sequences from the Hollywood movie actions dataset and the corresponding salient features. From left to right: an action frame, saliency map and the top 10% salient regions (used in feature computations).

action sequences or movie actions. In fact one choice may not work for all the different sequences. But as with many other methods, we try to determine the parameter choices that maximize the performance for each dataset individually. For the KTH and the Hollywood movie actions dataset, we determined the optimal choices using the training (and validation) set. For the UCF sports action dataset, 40 sequences ($\approx 35\%$ of the data) were used as a validation set for parameter selection. The best choices are listed in Table 7.1. We searched a range of spatial windows from 14×14 to 20×20 pixels, temporal windows from 4 frames per window to 10 frames per window, all in steps of 2. For each combination, we determined the top $10\%, 20\%$, and 30% salient regions and computed the corresponding feature descriptors in our bag-of-feature framework. We notice that the best choices do not vary by much between datasets. The best choices of patch sizes are more or less the same and they are also in agreement with other approaches [87]. There were only minor differences in the temporal window sizes between the different datasets. We believe that the reason for the UCF sports actions and Hollywood movie action sequences requiring more salient features in the model is because these actions (Golf swing, Answering phone, Getting out of car, etc.) required additional information about the scene and the interaction with the objects present in the scene for accurate classification.

Finally we also determine the size of the feature vocabulary to be used in the bag-of-features framework for each feature type. In most cases we got best results for a choice between 1500 and 3000 feature words. Smaller vocabularies (1500-2000 words)

Dataset	Patch size	Temporal window size	[%] Salient considered
KTH Actions	16×16	4	10
UCF Sports Actions	18×18	6	30
Hollywood movie actions	16×16	6	30

Table 7.1: Best parameter choices for each dataset determined through cross-validation on the training set.

produced best results for the region covariance descriptor, whereas larger vocabularies with 3000 words were more suited for HOG feature descriptors.

7.3.3 Results and Discussion

Using the optimal parameter choices discussed in the previous section, we train the SVM classifiers for each dataset. For the KTH and Hollywood movie actions dataset the training set and validation set were used to determine the best kernel choices for SVMs as well as their parameters (penalty factor, distance scaling variable, etc.). For the UCF sports action dataset 40 sequences representing all action classes were used

in cross-validation for determining the kernel parameters (penalty factor, distance scaling variable, etc.). For the KTH actions dataset best results were obtained using the Gaussian radial basis function, whereas for the UCF sports actions dataset as well as the Hollywood movie actions dataset, we obtained best performance with the exponential χ^2 kernel described previously. We compare the results of our approach using the HOG descriptor with other methods and we also report our performance with the region covariance descriptor. Due to the lack of availability of individual implementations of other feature detection methods (alone to be used with region covariance descriptor) we do not compare the results of using the covariance descriptor with other methods. Finally we combine the best region covariance descriptor based classifier and the best feature HOG classifier for each dataset as follows. We obtain the best bag-of-features histogram corresponding to each feature and then concatenate them together. This histogram is then renormalized and used as a combined feature [66] for classification. This method performed best for the Hollywood movie actions dataset and comparably for the other datasets. The results of performance for each dataset are shown in Tables 7.2, 7.3, and 7.4. Note that for the KTH and the UCF sports actions dataset, the accuracies corresponding to each feature descriptor are shown. However, for the Hollywood actions dataset the accuracies shown are for the combined feature method only as it performed the best. The corresponding accuracies shown for the method of Laptev *et al.* [74] are also results from the best feature combinations. Note that in our analysis

Feature Detector	HOG descriptor	Covariance descriptor	Combined descriptor
Harris 3D	80.9%	-	-
Cuboid	82.3%	-	-
Hessian	77.7%	-	-
Dense	79%	-	-
Our method	**85.3**%	**88.2**%	**90.1**%

Table 7.2: Average accuracies for different methods on the KTH actions dataset. Performance on the test set is shown.

we do not include any optical flow based feature descriptors. Even though these descriptors provide good performance, they are very expensive to compute. On the other hand the HOG and the covariance descriptors are relatively inexpensive. We improve the state-of-the-art in both the KTH actions and the UCF sports action dataset. We perform comparably with the approach of Laptev *et al.* in the Hollywood dataset. However, distinguishing between Sit-up and Sit-down, and Answering phone proved to be particularly challenging.

7.3.4 Time Taken for Computation

Our implementations were primarily carried out in MATLAB with some optimized components in C++. On average our method could process ≈ 0.5 frames per second while

Feature Detector	HOG descriptor	Covariance descriptor	Combined descriptor
Harris 3D	71.4%	-	-
Cuboid	72.7%	-	-
Hessian	66.0%	-	-
Dense	77.4%	-	-
Our method	**80.2**%	**85.92**%	**84.1**%

Table 7.3: Average accuracies for different methods on the UCF sports actions dataset. (Leave-one-out cross-validation).

computing one feature descriptor only, and ≈ 0.4 frames per second while computing both the HOG descriptors as well as region covariance descriptors. Since features are computed for each spatio-temporal window and not for each frame individually, we typically obtain a dense set of features. This is further influenced by the choice of the threshold for top $M\%$ saliency consideration. Table 7.5 shows processing rate comparisons for different methods obtained from [87] with our method.

Action	STIP Laptev et al.(CVPR '08)	Our Method
AnswerPhone	**32.1**%	17.1%
GetOutCar	**44.5**%	28.3%
HandShake	**32.3**%	31.1%
HugPerson	40.6%	**53.1**%
Kiss	53.3%	**60.2**%
SitDown	**38.6**%	26.3%
SitUp	**18.2**%	14.1%
StandUp	50.5%	**58.3**%
Overall	**38.38**%	36.1%

Table 7.4: Average accuracies for different methods on the Hollywood movie actions dataset. Performance on the test set is shown.

Feature Detector + Descriptor	Processing Speed (frames per second)
Harris 3D + HOG	1.6
Hessian + ESURF	4.6
Cuboid	0.9
Dense + HOG3D	0.8
Our method + HOG	0.5

Table 7.5: Average processing speed comparison for different feature detector and descriptor combinations.

7.4 Conclusion

To conclude, we summarize the contributions of this work. We propose a novel spatio-temporal feature detector based on the sparse representation length of the spatio-temporal patches measured via the residual error. We establish the theoretical motivation to determine spatio-temporal features in this manner. The patches are ranked according to their spatio-temporal saliency determined by the error magnitudes. These features are also determined in a multi-scale approach thereby making the feature robust to variation in spatial and temporal scales. We then compute the region covariance and HOG descriptors corresponding to the salient spatio-temporal patches. These features

are used in a bag-of-features approach with an SVM classifier framework. We improve upon the state-of-the-art in two (KTH actions and UCF sports actions) of the three datasets evaluated and we perform comparably on the third (Hollywood movie actions dataset). We obtain competitive performance without the computation of expensive features such as optical flow. Our method is computationally efficient and theoretically well founded.

Chapter 8

Conclusions and Future Work

We briefly summarize the contributions of this thesis as follows:

- We propose a novel reconstructive approach to measure saliencies in images. Specifically, this saliency score is global and does not measure saliency with respect to just local variations. Our approach is well founded in the theory of complexity and description length. We have also provided a solution to make this measure multi-scale.

- We use our proposed method for detecting salient objects and for predicting visual attention in images. While it was useful in both applications, it was more suited to foreground detection.

- Since our measure is based on self-similarity—*i.e.,* the representation error corresponds to how dissimilar each image region is with respect to the entire image—

we develop two efficient novel self-similarity descriptors which can be used in a classification framework to do object recognition.

- We also use our saliency detection method in a salient bag-of-features approach to enhance traditional methods of object classification.

- We also show the relevance of our measure in higher dimensions when we extend our approach to classifying human actions in spatio-temporal sequences. We are able to capture successfully the most salient motion characteristic of actions even amidst significant background variation.

8.1 Future Work

- Our method is based on sparse representation. However, dictionary learning and sparse coding are not the only ways to determine sparse representation error. Methods like Robust PCA [93] or Mean Shift clustering [94] can also be used to determine a representative basis for suitable domains. We neglected these approaches as they were just different tools to solve the same problem. Also, we had access to more efficient implementations for dictionary learning and sparse coding.

- We do not perform any post processing on our saliency maps as we use only the top $M\%$ salient regions for most applications. For the purpose of predicting visual attention, we can alternatively use some form of non-maximal suppression to reject

weak responses [95].

- We would like to improve our foreground detection method with traditional segmentation methods based on color and other cues to accurately delineate foreground objects.

- We are working on improving the robustness of our self-similarity descriptors by using a different representation space that is invariant to slight rotation and viewpoint changes.

- We can also extend segmentation to the spatio-temporal domain to determine the most salient motion patterns. This has a lot of uses in crowd behavior analysis, traffic pattern analysis, etc.

References

[1] M. Marszałek and C. Schmid, "Accurate object localization with shape masks," in *IEEE Conference on Computer Vision & Pattern Recognition*, June 2007.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *European Conference on Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[3] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[4] T.J. Smith, R.A. Henning, and K.U. Smith, "Performance of hybrid automated systems: a social cybernetic analysis," *International Journal of Human Factors in Manufacturing*, vol. 5, no. 1, pp. 29–51, 1995.

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,*. IEEE, 2006, vol. 2, pp. 2169–2178.

[6] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, New York, NY, USA, 2007, CIVR '07, pp. 401–408, ACM.

[7] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," *European Conference on Computer Vision–ECCV 2006*, pp. 589–600, 2006.

[8] G. Somasundaram, V. Morellas, and N. Papanikolopoulos, "Object classification with efficient global self-similarity descriptors based on sparse representations," in *IEEE Internation Conference on Image Processing (ICIP)*. IEEE, 2012.

[9] G. Somasundaram, Morellas. V, and Papanikolopoulos.N, "Action recognition using global spatio-temporal features derived from sparse representation," *Submitted to Computer Vision and Image Understanding.*

[10] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[11] M. C. Stone, "A survey of color for computer graphics," *SIGGRAPH*, 2001.

[12] T. Smith and J. Guild, "The CIE colorimetric standards and their use," *Transactions of the Optical Society*, vol. 33, no. 3, pp. 73, 2002.

[13] S. Nilufar, L. Chen, and H.K. Kwan, "A DLSI approach for content-based image classification," *2004 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, pp. 138–143, July 2004.

[14] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman, "Using latent semantic analysis to improve access to textual information," in *CHI '88: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1988, pp. 281–285, ACM.

[15] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.

[16] C. Harris and M. Stephens, "A combined corner and edge detection," in *Proceedings of The Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[17] J. Shi and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994*, June 1994, pp. 593 –600.

[18] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[19] S. Belongie and J. Malik, "Matching with shape contexts," *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries, 2000.*, pp. 20–26, 2000.

[20] D.G. Lowe, "Object recognition from local scale-invariant features," *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 vol.2, 1999.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, Eds., INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005, vol. 2, pp. 886–893.

[22] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 300–312, 2007.

[23] T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.

[24] R. Achanta, S. Hemami, F. Estrada, and S. Suesstrunk, "Frequency-tuned salient region detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[25] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07.* IEEE, 2007, pp. 1–8.

[26] N.D.B. Bruce and J.K. Tsotsos, "Saliency, attention, and visual search: An information theoretic approach," *Journal of Vision*, vol. 9, no. 3, 2009.

[27] D. Gao and N. Vasconcelos, "Bottom-up saliency is a discriminant process," in *IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–6.

[28] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," *Journal of American Mathematical Society*, vol. 22, no. 1, pp. 211–231, 2009.

[29] N. Vereshchagin and P. Vitányi, "Kolmogorov's structure functions with an application to the foundations of model selection," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002*. IEEE, 2002, pp. 751–760.

[30] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[31] K. Engan, S.O. Aase, and J.H. Husoy, "Frame based signal compression using method of optimal directions (MOD)," in *the Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, Jul 1999, vol. 4, pp. 1–4.

[32] S.G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[33] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *CS Technion*, 2008.

[34] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[35] Z. Wang and B. Li, "A two-stage approach to saliency detection in images," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.* IEEE, 2008, pp. 965–968.

[36] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 17–31, 2007.

[37] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[38] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," *Advances in neural information processing systems*, vol. 19, pp. 545, 2007.

[39] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 2376–2383.

[40] W.T. Freeman and E.H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[41] C.X. Ling, J. Huang, and H. Zhang, "Auc: a statistically consistent and more discriminating measure than accuracy," in *International Joint Conference on Artificial Intelligence*. Lawrence Erlbaum Associates LTD, 2003, vol. 18, pp. 519–526.

[42] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *IEEE Conference on Computer Vision and Pattern Recognition 2007 (CVPR'07)*, June 2007.

[43] T. Deselaers and V. Ferrari, "Global and efficient self-similarity for object classification and detection," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 1633–1640.

[44] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for multi-class object layout," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2010, pp. 229–236.

[45] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[46] J. Liebelt, C. Schmid, and K. Schertler, "Viewpoint-independent object class detection using 3D feature maps," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008.* IEEE, 2008, pp. 1–8.

[47] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[48] B.E. Trevor, T. Hastie, L. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2002.

[49] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *CS Technion*, 2008.

[50] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning.* ACM, 2009, pp. 689–696.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[52] T.W. Lee and M.S. Lewicki, "Unsupervised image classification, segmentation, and enhancement using ICA mixture models," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 270–279, 2002.

[53] M. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in unsupervised image classification," in *Fifth World Enformatika Conference (ICCI), Prague, Czech Republic.* Citeseer, 2005, pp. 199–204.

[54] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *in the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, June 2006.

[55] G. Somasundaram, V. Morellas, N. Papanikolopoulos, and L. Austin, "Counting pedestrians and bicycles in traffic scenes," in *the Proceedings of the IEEE Intelligent Transportation Systems Conference, St. Louis*, 2009.

[56] R. Sivalingam, G. Somasundaram, V. Morellas, V. Papanikolopoulos, O. Lotfallah, and Y. Park, "Dictionary learning based object detection and counting in traffic scenes," in *the Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras, Atlanta*, 2010.

[57] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[58] C. Cortes and V. Vapnik, "Support vector networks," in *Machine Learning*, 1995, pp. 273–297.

[59] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, 2008.

[60] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[61] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for multi-class object layout," in *the Proceedings of the International Conference on Computer Vision (ICCV) Kyoto, Japan*, 2009.

[62] C. Wang, D. Blei, and L. Fei-Fei, "Simultaneous image classification and annotation," in *the Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.

[63] L. J. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.

[64] B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett, "New support vector algorithms," *Neural Computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[65] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27,

2011, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[66] N. Larios, H. Deng, W. Zhang, M. Sarpola, J. Yuen, R. Paasch, A. Moldenke, D.A. Lytle, S.R. Correa, E.N. Mortensen, et al., "Automated insect identification through concatenated histograms of local appearance features: feature vector generation and region detection for deformable objects," *Machine Vision and Applications*, vol. 19, no. 2, pp. 105–123, 2008.

[67] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1999, vol. 2.

[68] O. Masoud and N.P. Papanikolopoulos, "Using geometric primitives to calibrate traffic scenes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2004, vol. 2, pp. 1878–1883.

[69] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," Tech. Rep., Dept. of Computer Science, Technion, April 2008.

[70] Y. Malinovskiy, J. Zheng, and Y. Wang, "Model-free video detection and tracking of pedestrians and bicyclists," *Computer-Aided Civil and Infrastructure Engineering*, vol. 24, no. 3, pp. 157–168, 2009.

[71] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1713 –1727, October 2008.

[72] D. Fehr, R. Sivalingam, V. Morellas, N. Papanikolopoulos, O. Lotfallah, and Y. Park, "Counting people in groups," in *the Proceedings of the Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2-4 2009, pp. 152 –157.

[73] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[74] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.

[75] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *VS-PETS*, 2005, pp. 65–72.

[76] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004.* IEEE, 2004, vol. 3, pp. 32–36.

[77] J.K. Aggarwal and M.S. Ryoo, "Human activity analysis: A review," in *To Appear in ACM Computing Survey.*

[78] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden markov models," *IEEE Transactions on Image Processing*, vol. 2005.

[79] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, December 2007.

[80] K. Guo, P. Ishwar, and J. Konrad, "Action recognition using sparse representation on covariance manifolds of optical flow," in *2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2010, pp. 188–195.

[81] K. Guo, P. Ishwar, and J. Konrad, "Action recognition in video by covariance matching of silhouette tunnels," *2009 XXII Brazilian Symposium on Computer Graphics and Image Processing*, pp. 299–306, 2009.

[82] A. Castrodad and G. Sapiro, "Sparse modeling of human actions from motion imagery," Tech. Rep., DTIC Document, 2011.

[83] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," *European Conference on Computer Vision–ECCV 2008*, pp. 650–663, 2008.

[84] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011.

[85] Q.V. Le, W.Y. Zou, S.Y. Yeung, and A.Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 3361–3368.

[86] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 3501–3508.

[87] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *University of Central Florida, U.S.A*, 2009.

[88] L. Shao and R. Mattivi, "Feature detector and descriptor evaluation in human action recognition," in *Proceedings of the ACM International Conference on Image and Video Retrieval*. ACM, 2010, pp. 477–484.

[89] I. Sipiran and B. Bustos, "Harris 3D: a robust extension of the harris operator for interest point detection on 3D meshes," *The Visual Computer*, pp. 1–14, 2011.

[90] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic Resonance in Medicine*, vol. 56, no. 2, pp. 411–421, 2006.

[91] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," in *Conference on Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06.* IEEE, 2006, pp. 13–13.

[92] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach: A spatio-temporal maximum average correlation height filter for action recognition," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[93] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *Analysis*, p. 23, 2010.

[94] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[95] C. Sun and P. Vallotton, "Fast linear feature detection using multiple directional non-maximum suppression," *Pattern Recognition*, vol. 2, pp. 288–291, 2006.