USC-SIPI REPORT #419

Biologically Inspired Overcomplete Representation, Feature Extraction and Object Classification

by

Pankaj Mishra

August 2011

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Viterbi School of Engineering
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Suite 400
Los Angeles, CA 90089-2564 U.S.A.

BIOLOGICALLY INSPIRED OVERCOMPLETE REPRESENTATION, FEATURE EXTRACTION AND OBJECT CLASSIFICATION

by

Pankaj Mishra

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

August 2011

To My Parents and Sister

Acknowledgements

I would like to thank my adviser Prof. B. Keith Jenkins for his guidance, support and patience. His experience, attention to details and knowledge has significantly contributed to my academic growth during my graduate studies.

I also thank my committee members Prof. Alexander A. Sawchuk and Prof. Stefan Schaal for their useful comments and suggestions. There are many other people who have contributed directly and indirectly to this thesis. I would like to express my gratitude to Abinav Sethy, Amol Bakshi, Zack Baker, Jason Dziegielewski, Abhisek and Ponghkhi Borah, Iain Bailey, Kristopher Ian Stevens, Jen and Patrick Gorman, Sabyasachi Ghosh, Tarun Jain, Kimish Patel, Kalpesh Solanki, Kartik Audhkhasi and Christiann Boutwell for making my graduate stay enjoyable.

Table of Contents

Dedication ii

Acknowledgements iii

List of Figures vi

List of Tables viii

Abstract ix

Chapter 1 Introduction 1
- 1.1 Objective . . . 1
- 1.2 Challenges . . . 2
- 1.3 Our approach . . . 2
- 1.4 Contribution of this research . . . 4
- 1.5 Outline of thesis . . . 7

Chapter 2 Overcomplete Basis Functions 8
- 2.1 Mathematical model . . . 9
  - 2.1.1 Calculating Coefficients . . . 10
  - 2.1.2 Calculating Basis Functions . . . 12
- 2.2 Learning . . . 13
  - 2.2.1 Random seed based basis function . . . 15
  - 2.2.2 Result and Discussion . . . 16
  - 2.2.3 Wavelet coefficients based basis function . . . 21
- 2.3 Measuring Sparseness . . . 21
  - 2.3.1 Kurtosis . . . 23
  - 2.3.2 Modified Treves-Rolls . . . 23
  - 2.3.3 Activity Sparseness . . . 25
- 2.4 Advantages of Sparse Coding . . . 25

Chapter 3 Feature Extraction Models 27
3.1 Modified HMAX based C2 features . . . 27
3.1.1 Pre-S1 Layer . . . 28
3.1.2 Simple Layer S1 . . . 28
3.1.3 Complex Layer C1 . . . 29
3.1.4 Simple Layer S2 . . . 29
3.1.5 Complex Layer C2 . . . 30
3.1.6 Dictionary of S2 prototype patches . . . 30
3.2 Global feature . . . 32
3.2.1 Modified Gist based global features . . . 33
3.2.2 Spatial pyramid based global features . . . 35
3.3 Mid-level features based Visual perception . . . 36

Chapter 4 Feature Combination Methods 38
4.1 Classifier combination strategies . . . 39
4.2 Single- vs. Multiple- Kernel Learning . . . 41

Chapter 5 Datasets and Experiments 43
5.1 Dataset . . . 43
5.1.1 Caltech-101 . . . 43
5.1.2 Caltech-5 dataset . . . 45
5.1.3 Oxford Flowers dataset . . . 45
5.2 Experimental Setup . . . 47
5.2.1 Caltech-101 dataset . . . 47
5.2.2 Caltech-5 dataset . . . 48
5.2.3 Oxford Flower dataset . . . 48

Chapter 6 Results 49
6.1 Caltech-5 . . . 49
6.2 Caltech-101 . . . 51
6.2.1 Comparison of Results: Caltech-101 . . . 55
6.3 Oxford Flowers . . . 56
6.3.1 Comparison of Results: Oxford Flowers . . . 58

Chapter 7 Conclusion 60
7.1 Future Work . . . 61

References 63

List of Figures

2.1 Three different sparseness terms and their derivatives plotted w.r.t coefficient a_i. The dash-dot red curve is the original [47] sparseness term, while the solid blue curve is our approximate L1-norm term and the dashed black curve is the sparseness term based on [58] 10

2.2 Steps involved in generating the natural-stimuli adapted basis functions from randomly sampled images patches taken form [21] 15

2.3 Preprocessed training images, left column has 3 of the natural images & right column has 3 of man-made structure images 18

2.4 Random samples from natural scene images 19

2.5 Random samples from man-made structure images 19

2.6 Basis functions for tree foliage 20

2.7 Basis functions for Man-made structures 20

2.8 Natural-stimuli adapted filters and their spectra produced using a modification of [47, 48]. 22

2.9 Lifetime sparseness measured over tree foliage dataset. “Initial” here means at the start of the iteration while “final” means the sparseness measure after stimuli was presented. The coefficients learnt were initialized randomly. Please, note that x-axis denotes the number of iterations while y-axis denotes the three sparseness measure mentioned in sec. 2.3. 24

3.1 The response of different layers of modified HMAX model. In (a) the first image is the original image and the next image is the grayscale, resized, and locally contrast normalized version of it. In (b) the four orientation images of one resolution level of the S1 layer images are shown. In (c) the C1 features of the model for one resolution layer are shown. In (d) the responses of the S2 layer for 4 dictionary patches are shown, again for only one resolution layer. 31

3.2 An example of typical modified gist features for 4 orientations and one scale. The original image is the first image shown in Fig. 3.4. 33

3.3 The spatial layout of the images for carrying out muiltscale feature extraction [33]. 35

3.4 A typical set of mid-level features based on the Gestalt principle of continuity in visual perception [5]. The first image is the grayscale version of the original image. The next five images show the Gestalt features for different resolutions. The pseudocolor denotes the edge orientation and the color saturation denotes the relative strength of edges. 37

5.1 Five randomly selected images from each of 24 of the image categories (picked randomly) of Caltech-101. Overall there are a total of 102 categories in this dataset. 44

5.2 Few of the typical images in the Oxford flower dataset. Overall there are a total of 17 categories in this dataset. 46

6.1 Precision-Recall curve for Caltech-5 dataset. AUC stands for area under the curve. 50

6.2 Comparison of different combinations of four feature sets, using the arithmetic mean combiner. C2: modified HMAX C2 features; GistC: modified color gist features; Gest: features based on Gestalt principle of continuity; SP: Spatial pyramid based features. The results are based on an average of 10 independent runs for both 15/50 and 30/50 cases. 54

6.3 Comparison of different combinations of four feature sets, using the arithmetic mean combiner. The results are based on an average of 5 independent runs. 58

List of Tables

6.1 Comparison (% correct classification) of our results for ($object|\overline{object}$) with the benchmark [17, 70]. 49

6.2 Performance (percent correct) of individual classifiers. C2: modified HMAX C2 features; Gest: features based on Gestalt principle of continuity; GistC: modified color gist features; SP: Spatial pyramid based features. Each column heading gives the number of training and testing images. The results are based on an average of 10 independent runs for 15/50 case and 10 independent runs for 30/50 cases. Here 15 and 30 are the number of training images and 50 is the number of test images used. 51

6.3 Comparison of classification performance based on combined feature sets; percent correct based on different fusion methods. All four feature sets were used. The results are based on an average of 10 independent runs for both 15/50 and 30/50 cases. 52

6.4 Comparison of classification results (percent correct) for 5 most difficult classes over individual feature sets and their overall combination. The results are based on an average of 10 independent runs for 15/50. 53

6.5 Comparison of classification results (percent correct) for 5 highest performing classes over individual feature sets and their overall combination. The results are based on an average of 10 independent runs for 15/50. 53

6.6 Performance of individual classifiers for Oxford Flowers dataset. The column heading gives the number of training and testing images.The results are based on an average of 5 independent runs. 57

6.7 Comparison of classification performance for Oxford Flowers dataset based on combined feature sets; percent correct based on different fusion methods. All four feature sets were used. The results are based on an average of 5 independent runs. 57

Abstract

A key to solving the multiclass object recognition problem is to extract a set of features which accurately and uniquely capture the salient characteristics of different objects. We show that complementary kinds of feature sets *e.g.*, based on local, mid-level and global characteristics, can be combined to significantly improve recognition accuracy over that obtained using individual (or subcombination of) feature sets.

First, we extract a set of local features based on a modified HMAX model, which is a hierarchical computational framework inspired by mammalian visual cortex. One of our modifications uses natural-stimuli adapted filters in place of Gabor filters. Overcomplete sets of basis functions based on sparseness maximization criteria have been reported to closely mimic the mammalian visual cortex, V1, in the sense that the resulting basis functions are typically localized, oriented, and bandpass, as are filters in V1. These overcomplete basis functions allow a smooth transition of coefficients and allow a high degree of specificity to image statistics. These natural-stimuli adapted filters are used with the HMAX model which increases its biological plausibility. The resulting features are largely scale, translation and rotation invariant. Second, we extract contextual information using modified Gist and spatial pyramid based features. Third, to capture larger contours and edges we extract features based on the Gestalt principle of continuity in visual perception.

We combine these feature sets using confidence measures derived from discriminative-model based posterior probabilities. Each posterior probability obtained in our case is based on support vector machine (SVM) decision boundaries, in part because SVMs have been shown to do well on large datasets. Different combinations of confidence measures are explored. Most significant improvements are gained using non-trainable fusion techniques. We demonstrate significant improvement for object recognition performance (over individual feature sets) using the publicly available Caltech-101 and 17-species Oxford Flowers datasets. The progressive addition of feature sets always resulted in performance improvement though the incremental gains varied.

Chapter 1

Introduction

1.1 Objective

One of the most important problems in machine vision is to be able to recognize different kinds of objects in various backgrounds. The generic object recognition problem becomes even more challenging as one has to take into account a large number of variations due to factors like object position, view point, illumination, clutter and object deformation [9, 52, 53]. This object recognition task attains even more importance in the current digital age given the ease with which images can be captured, stored and processed.

In this thesis we tackle the object recognition problem by extracting useful features which are then used to carry out multiclass object classification. This approach is inspired from most of the successful computational models [7, 13] for object recognition, which take a two step approach: an "appropriate" feature extraction step followed by a classification step. The key challenge for the extracted features is that they should be mostly scale, translation and rotation invariant. At the same time the feature extraction step should extract the features in such a manner that the features are relatively invariant to intraclass variation but capture enough information for interclass discrimination. For the classification step, to obtain good classification results the classifier should be able to maintain a trade-off between discriminative power and invariance [7, 13, 67]. Towards this goal the most successful methods to date have been based on kernel methods [42, 61] or on a combination of weak classifiers [63, 64].

1.2 Challenges

The feature extraction methods towards solving object recognition problems have certain key characteristics, advantages and disadvantages. Current feature extraction methods can be broadly divided into patch based and histogram based methods. One of the most successful, the bag-of-words method [10, 15], extracts features from different parts of images using local invariant features, *e.g.*, SIFT features [15, 42]. The different patches from the images are selected randomly and are assumed to have equal probability. This model despite its simplistic assumption performs very well on object recognition problems. Models which try to keep the relative geometrical information among different patches, e.g., the parts-and-structure model [17], have also been demonstrated to be quite successful on the object recognition problem, but come with additional computational cost. Most of these models employ a dictionary of words based approach and these dictionaries are extracted either from the given dataset or are universal [60] *i.e.*, learnt from independent data.

The other popular approach uses histograms of the edges of the underlying images and is known as the histogram-of-gradient approach. These models and their hierarchical versions have been shown to perform well [8, 33]. References [5, 60] point out that whereas bag-of-words methods might not work well under geometric transformations, histogram based methods might be too specific for generic object recognition.

1.3 Our approach

In this work we look into a set of feature extraction methods and their possible combination towards solving the multiclass object recognition problem. First we start with an HMAX [55, 61] model, which tries to find a balance between invariance and selectivity of the image

representation via a set of features. This model which is inspired by biology uses a set of alternating simple- and complex- cell layers to extract largely scale, translation and rotation invariant features. We modify this HMAX model by further incorporating the underlying statistics of images; the biological plausibility of the HAMX model is extended via a set of natural-stimuli adapted filters [39, 47]. The second set of features we examine is based on one of the Gestalt principles of visual perception. We use continuity based intermediate features [6] which combine short edges to capture longer edges and thus possibly the shape of the objects.

Both HMAX based and Gestalt based feature sets take an object centric view of the feature extraction process. In the third and fourth sets of features we follow a more holistic approach which takes context of the objects in to account. Under this viewpoint we first look into a set of Gist features [45, 64] which incorporate context by taking average energy of features over a set of non-overlapping windows. The contextual information helps in case of object ambiguity due to image degradation in the presence of occlusion, illumination, shadows etc. resulting in poor resolution [45]. The final set or second global set of features we use are the spatial pyramid based features introduced in [33]. These features are extracted by progressively dividing the given scene into sub-regions and then calculating the weighted histograms of the features for each of these sub-regions.

To utilize these different sets of features for the object recognition task different methods have been suggested [14, 29, 31]. For example in [6, 41] features are combined before they are fed to the classifiers. In our work we combine them using the nontrainable fusion method suggested in [29, 31]. As the four sets of features obtained present different aspects of the underlying dataset, we adopt the strategy of using posterior probability based confidence measures to combine the features.

1.4 Contribution of this research

This work which combines different feature sets towards multi-class object recognition has the following novel contributions:

- To generate the sparse overcomplete representation, Olshausen *et al.* proposed a cost optimization method that uses an L2-norm based regularization term. We extend this model to include approximate-L1 norm based regularization term to produce localized, oriented and bandpass filters [46] similar to ones found in mammalian visual cortex. The approximate-L1 norm based regularization term results in better sparseness than L2-norm based term.

- The final outcome of overcomplete representation is influenced by the initial conditions. We experimented with both single scale as in [48] and multi-resolution basis functions as in [49]. For the single scale basis functions we carried out the experimented on both “natural” and “man-made structure” images. Some of the key characteristics of the resulting basis functions are also discussed.

- One of the key assumptions in the HMAX model is that the first layer of simple cells uses Gabor filters to extract features. This assumption which is very robust depends on set mathematical formula and its parameters, but doesnt take into account characteristics of natural images per se.

  The HMAX model was proposed in a bid to understand higher level vision tasks such as multiclass classification and object recognition. Hence, the HMAX model modified to incorporate these natural-stimuli adapted filters becomes a suitable candidate for feature extraction and object recognition. In this work we extend the

HMAX model to include natural-stimuli based filters. The object recognition capability of this modified HMAX model is demonstrated by applying it to multi-class object recognition task on two publicly available Caltech-101 [15] and Oxford-17 [44] flower datasets.

- For our modified HAMX model we experimented with two separate methods a) K-means based and b) L2-norm based to prune the dictionary. In our case we were able to use a comparatively smaller dictionary to produce similar results.

- Gainfully combining the different feature sets for an object recognition problem can be done in a variety of ways. If the features are combined before they are fed to the classifiers, the different features can be concatenated into one vector [6] or they can be modelled probabilistically e.g., using mixture-of-Gaussians. On the other hand, if each set of features is first classified and the estimates for classification results are then combined, we can use a fixed combining rule [14], a logistic regression model [63], or a boosting based model [57].

  In the present effort we first classify each set of features independently, and then use a fixed combining rule to combine the estimates based on the individual features. The use of this non-trainable combiner, which according to [14, 29] is a fixed combining rule that allows the ensemble to be used as soon as the base classifiers have been trained. The different features were combined by weighted average based on cross validation before assigning the final class labels. Our results indicate that if appropriate qualitatively different sets of features are properly combined then they can be used to advantage in solving multiclass classification problems. We demonstrate that appropriate features extracted using different methods can be combined to provide

better results on multiclass object recognition than with features extracted using any one of the methods.

- We experimented with 4 different non-trainable combiners based on Product, Geometric, Harmonic mean and Maximum rules as suggested in [14, 29, 31]. The arithmetic mean based combiner gave us the best result; it slightly outperformed the product based combiner. Though contrary to observation made in [30, 31] we found that product based combiner is stabler (as in lower standard deviation) than arithmetic mean based combiner.

- In this work we used posterior probability of different classifiers as a confidence measure for combination as suggested in [29, 31]. To calculate the posterior probabilities for different classifiers we experimented with linear kernel as well a linear combination of linear, Gaussian and polynomial kernels as in multiple kernel learning (MKL) framework [2, 66, 67] based support vector machines. Based on our experiments we found that both the methods produce similar results and due to its lower computational complexity we used linear kernel only based support vector machine (SVM) for final results.

- To capture context based appearance features Torralba *et al.* [45, 64] proposed intensity based gist features. We extend these features to include color-opponecy based RG and BY channel to include more information than intensity based channel only. Our results for gist features confirm that inclusion of color channels improve classification result vs. using only intensity channels.

1.5 Outline of thesis

This thesis is structured as follows. In Chapter 2 we describe ovecomplete basis function. In this chapter we first describe the model leading up to a sparseness-maximization based cost function. Thereafter we describe development of basis functions from random seeds and from wavelet-coefficient seeds. Then we describe some of the sparseness measure criteria along with some arguments in favor of sparse coefficients. In Chapter 3 we describe three types of feature extraction models which capture the local, mid-level and global characteristics of an underlying dataset. Under the local features model we describe our modification based on overcomplete basis functions. We argue the importance of mid-level and global features and describe their modified models. In Chapter 4 we first describe some feature and classifier combination strategies. Then we in detail describe the class-conscious based multiple classifier system we adapt for our work. In Chapter 5 we describe the three datasets namely Caltech-5, Caltech-101 and Oxford flowers dataset used in this work. We discuss some of the key aspects of these datasets. After that we describe the experimental setup we used to evaluate our feature extraction and classification system. In Chapter 6 we discuss our results and present various key evaluations of our results. Then we compare our results with similar works in the literature. In Chapter 7 we present some closing remarks along with some of the directions in which this work can be extended.

Chapter 2

Overcomplete Basis Functions

Mammalian visual systems have evolved over millions of years to optimally represent natural stimuli by efficiently utilizing the limited resources at their disposal. Primary visual cortex V1, one of the best studied parts of the visual system, has been shown to possess a high degree of specificity to natural stimuli. Generative models which try to learn overcomplete sets of basis functions based on sparseness maximization criteria have been reported [49] to closely mimic the mammalian visual cortex, V1, in the sense that the resulting basis functions are typically localized, oriented, and bandpass, as are filters in V1. These overcomplete basis functions allow a smooth transition of coefficients and allow a high degree specificity to image statistics [3, 20].

In this work we propose that the basis functions learned in this manner can be successfully applied to the problem of pattern recognition, specifically to the feature extraction and thus object recognition. The filters emerge as a result of optimization based in part on smooth L1-norm based sparseness maximization. The optimization method utilizes the underlying statistics of natural images to produce a set of sparse localized, oriented and bandpass filters. These resulting filters, which are a set of edge and line detectors [4], can be used to carry out the initial stages of feature extraction. We further use wavelet like coefficients to generate the multiscale version of the basis function as suggested in [49].

2.1 Mathematical model

Images can be expressed as a linear superposition of basis function and additive white Gaussian noise (AWGN), ν :

$$I(\vec{x}) = \sum_{i=1}^{M} a_i \phi_i(\vec{x}) + \nu(\vec{x}) \tag{2.1}$$

in which

$I(\vec{x})$ is the image vector, $(N \times 1)$

$\phi_i(\vec{x})$ is a basis vector, $(N \times 1)$

a_i is a coefficient (scalar),

In matrix notation: $I = \Phi A + \nu$,

in which

I is the image vector, $(N \times 1)$

Φ is a basis matrix, $(N \times M)$

A is a coefficient vector, $(M \times 1)$

For an overcomplete representation generally M > N, i.e the number of basis functions is greater than the dimensionality of the image. So, according to this model we have to find two variables:

1. A set of sparse and statistically independent coefficients.
2. A set of basis functions to represent the given image.

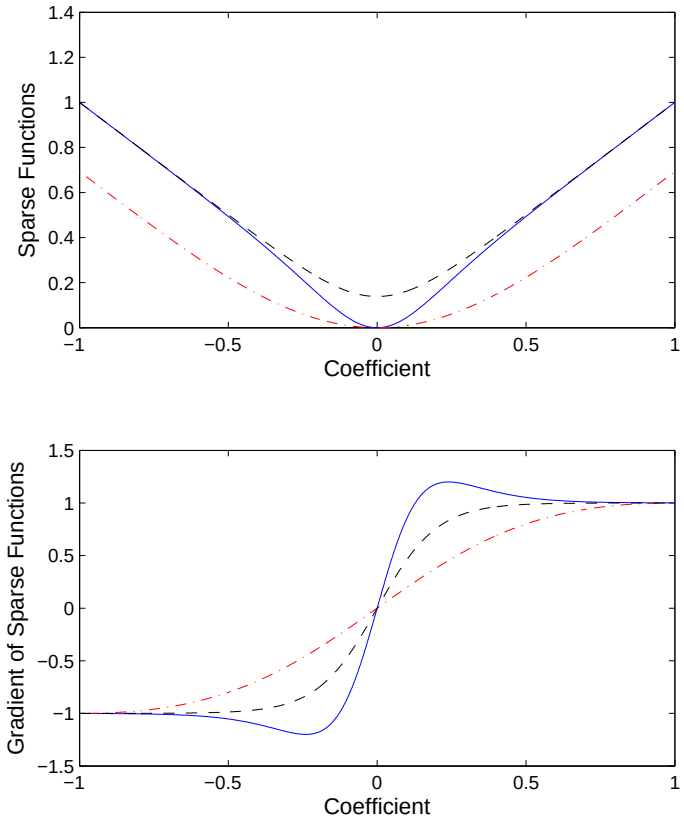


Figure 2.1: Three different sparseness terms and their derivatives plotted w.r.t coefficient a_i. The dash-dot red curve is the original [47] sparseness term, while the solid blue curve is our approximate L1-norm term and the dashed black curve is the sparseness term based on [58]

2.1.1 Calculating Coefficients

Using Bayes' rule, the posterior probability of coefficients $P(A|I,\Phi)$, given images (I) and basis functions (Φ), can be expressed in terms of likelihood and priors:

$$P(A|I,\Phi) \propto P(I|A,\Phi)P(A)$$

In the above equation $P(A|\Phi)$ has been approximated by P(A), as the probability distribution of coefficients is assumed to be dependent on the underlying image statistics, not on the set of basis functions.

The likelihood term is determined by the Gaussian noise, hence $P(I|A,\Phi) \propto \exp\frac{-|I-\Phi A|^2}{2\sigma^2}$, where σ^2 is the variance of the noise. The log of the posterior distribution is:

$$\log P(A|I,\Phi) \;=\; \left\{\frac{-1}{2\sigma^2}|I-\Phi A|^2\right\} + \log P(A) + const.$$

Maximizing this distribution will yield:

$$\hat{A} = \arg\max_A \left\{\frac{-|I-\Phi A|^2}{2\sigma^2} + \log P(A)\right\} \tag{2.2}$$

Natural images can be thought of as made from a number of independent "events" or "objects". This view can be included by assuming independence of coefficients. Hence, eq. (2.2) becomes:

$$\hat{A} = \arg\max_A \left\{\frac{-1}{2\sigma^2}|I-\Phi A|^2 + \log\prod_i P(a_i)\right\} \tag{2.3}$$

$$\tag{2.4}$$

$$= \arg\max_A \left\{\frac{-1}{2\sigma^2}|I-\Phi A|^2 + \sum_i \log P(a_i)\right\} \tag{2.5}$$

Now, Olshausen and Field suggested to parameterize the prior probability by $\exp\{-S(a_i)\}$. So, choosing $S(a_i)$ to be $\log(1+a_i^2)$ means the prior would have a Cauchy distribution. We use instead an approximate L1- norm as the sparseness term:

$$S(a_i) = \frac{a_i[1+\exp(-ca_i)]}{1-\exp(-ca_i)} \tag{2.6}$$

where c is a constant term which governs the accuracy of the L1-norm approximation. This approximation makes the sparseness term differentiable. The comparison of our sparseness measure with two other sparseness measures in Fig. 2.1 indicates its improved sparseness, especially for coefficient values close to zero. (Also note that the derivative for the original sparseness function almost vanishes for values close to zero, which might result in poorer sparseness, similar to the case of L2-norm [47].)

2.1.2 Calculating Basis Functions

The second goal is to adapt the functions over the given set of coefficients maximizing the log probability w.r.t. Φ :

$$\Delta\,\Phi \propto \frac{\partial}{\partial \Phi}\{\log P(I|\Phi)\}$$

$$= \frac{1}{P(I|\Phi)}\ \frac{\partial}{\partial \Phi}\int P(I|\Phi, A)P(A)dA$$

$$= \frac{1}{P(I|\Phi)}\ \int \frac{\partial}{\partial \Phi}\exp\left\{\frac{-|I-\Phi A|^2}{2\sigma^2}\right\}P(A)dA$$

Ignoring constants,

$$= \frac{1}{P(I|\Phi)}\ \int (I-\Phi A)A^T\ \exp\left\{\frac{-|I-\Phi A|^2}{2\sigma^2}\right\}P(A)dA$$

$$= \int (I-\Phi A)A^T\ \frac{P(I|\Phi, A)}{P(I|\Phi)}P(A)dA$$

$$= \int (I-\Phi A)A^T\ P(A|I, \Phi)dA \qquad (2.7)$$

In reaching the last step $P(A|\Phi)$ has been approximated by P(A), as the probability distribution of coefficients is assumed to be dependent on the underlying image statistics and is independent of the set of basis functions.

To calculate the $P(I|\Phi)$ we need to integrate the distribution over all the values of coefficients. This problem can be solved via two possible approximations. First, as suggested in [36, 38], solving the integral by using best-fitting Gaussian distribution. Second, as suggested by Olshausen and Field [47, 49], coefficients were sampled at their peak value. The problem with this approach is that coefficients peak around zero, which means the L2 norm of the basis can grow to infinity. This can be fixed by adding one more step of adjusting the norm of basis functions, to keep them from growing infinitely. (Refer to Olshausen and Field [49] for more details).

The equation for updating the basis becomes:

$$\triangle \Phi \propto (I - \Phi A)A^T \tag{2.8}$$

2.2 Learning

In the last sec. 2.1 we showed that the model for learning sparse structure of natural images leads to an optimization problem. So, the learning problem is determined by minimizing the following cost function:

$$E = \sum_{\vec{x}} \left[I(\vec{x}) - \sum_i a_i \phi_i(\vec{x}) \right]^2 + \lambda \sum_i \frac{a_i[1 + \exp(-ca_i)]}{1 - \exp(-ca_i)} \tag{2.9}$$

containing two terms:

1. Mean squared error term

2. Sparseness term.

This cost function (eq.2.9) tries to find a sparse structure (second term) while keeping the reconstructed image within an acceptable level of accuracy (first term). The relative weight of the two terms is decided by the value of the Lagrangian coefficient λ. The sparseness term learns coefficients by differentially suppressing the small values. It can be better illustrated by looking at the derivative of the sparseness term eq. 2.6. Following cite [47], from figure 2.1 it can be seen that the derivative of the second term is linear within a small range of a_i and non-linear outside. So, the coefficients a_i with small magnitude are suppressed as the learning progresses, while the coefficients of significant values are preserved.

The learning process is accomplished in two steps. In the first step the coefficients are learnt and in the second step the bases are updated so as to better approximate the original image.

1. Inner loop minimization : In the inner loop coefficients are learnt w.r.t. a_i while keeping basis functions ϕ constant. That is coefficients are learnt according to the solution of the eq. 2.5. As mentioned in the sec. 2.1, learning coefficients is governed by the sum of the weighted residue image and the sparseness criterion.

2. Outer loop minimization : In this step the basis functions are updated (keeping the coefficients constant) according to eq. 2.8, expressed in the following form:

$$\triangle \Phi = \eta \, (I - \Phi A) A^T \tag{2.10}$$

Here, η is learning rate. To expedite convergence the value of η can be varied after every few hundred iterations.

We experimented with two types of natural-stimuli adapted filters. These two types were based on initial seeding of the coefficients. They are described in the following three sections.

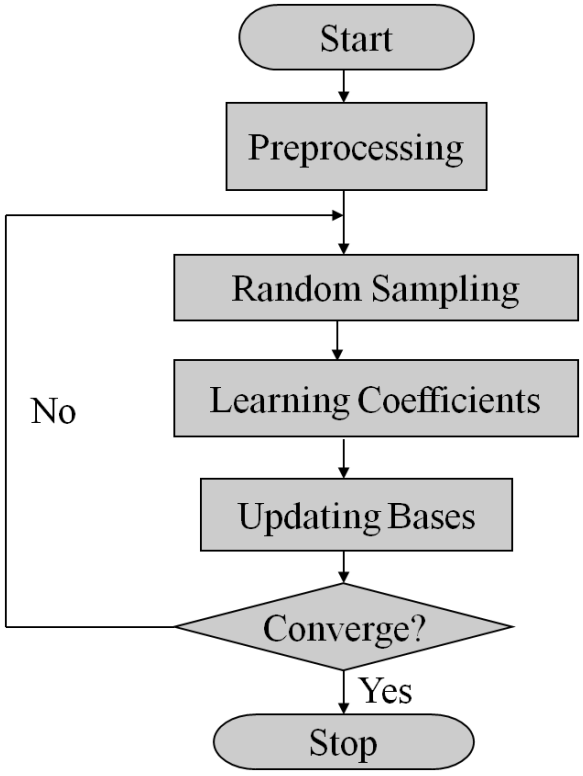


Figure 2.2: Steps involved in generating the natural-stimuli adapted basis functions from randomly sampled images patches taken form [21]

2.2.1 Random seed based basis function

The data set consists of 10 images of size 1024×1536. These images were taken from the data set used by van Haternen and van der Schaaf [21] for conducting biological verification of the data. The original data in [21] were stored as 12 bits per pixel. For practical

purposes such as speed and storage space, images were downsampled using cubic interpolation available in Matlab to 252×372.

The simulation process basically consists of the following steps:

1. Preprocessing
2. Sampling
3. Minimizing coefficients
4. Updating Basis functions

The flowchart of these steps leading to generation of natural-stimuli adapted filters is shown in fig. 2.2.

2.2.2 Result and Discussion

The experiment described in previous section was conducted on two sets of data. The first set of data comprised 10 different natural scenes (*e.g.*, tree foliage), while, the second set of data comprised 10 different man-made images structures (*e.g.*, buildings, streets). To learn the basis function images patches of various sizes are sampled from randomly selected location in the images. A few of these preprocessed images are shown in figure 2.3. Some of the randomly sampled 12×12 image patches used for simulation are shown in figure 2.4 and figure 2.5. A stable solution emerges after 5000 iterations, where each iteration comprises of learning coefficients followed by updating the basis functions. The resulting overcomplete set of basis functions for the natural data set is shown in figure 2.6. For the man-made structure data set, the resulting basis functions are shown in figure 2.7.

Most of the basis functions show features similar to Gabor wavelets [37, 46]. It has been shown [11, 18, 21] that these wavelet type basis functions seem to closely mimic the

receptive field of visual cortex, V1, which were reported by Hubel and Wiesel [25] to have "localized, oriented and bandpass" characteristics. Olshausen and Field [47] showed that the number of basis functions in higher spatial-frequency bands outnumbered the set of basis functions in lower spatial-frequency bands.

The set of basis functions highlight underlying structure of the training data. Basis functions from man made images have two predominant directions that are approximately orthogonal to each other. On the other hand, the set of basis functions from natural scenes are more varied and don't have preference for any particular direction. Also, features within man made basis functions are more elongated than features within natural scene basis functions. One point to be noted is that the length of features within diagonal basis functions are generally bigger than features other basis functions, possibly because of square image patches being used instead of circular patches.

As postulated by Field [18], these coefficients would have higher Kurtosis, as confirmed below in figure 2.9. One important thing to be kept in mind is that spareness-maximization doesn't guarantee a factorial (independent) code. Since the set of basis functions are over-complete and mostly singular hence not invertible. This is strikingly different from the basis functions obtained by other schemes (*e.g.,* ICA, PCA) wherein certain de-correlation properties were imposed and the coefficients can be inverted (*i.e.,* the coefficients can be calculated easily).

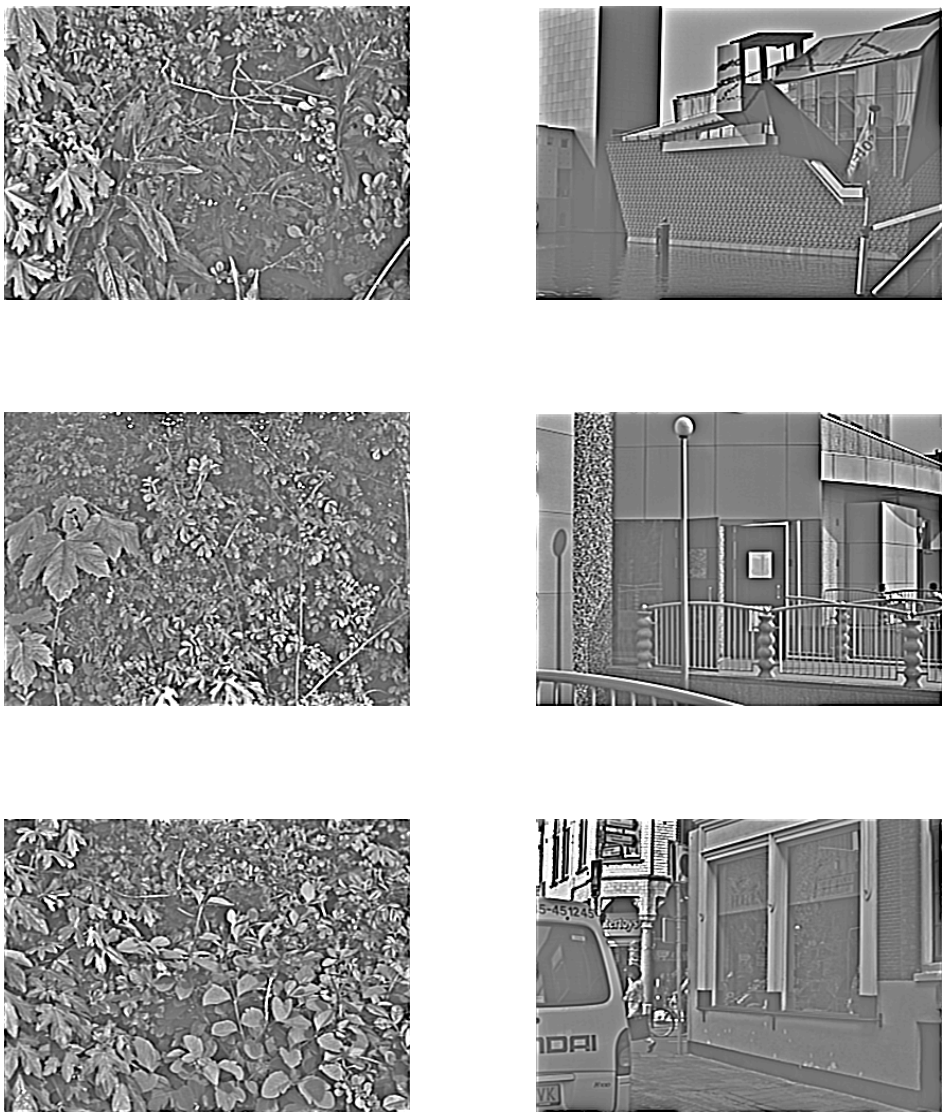


Figure 2.3: Preprocessed training images, left column has 3 of the natural images & right column has 3 of man-made structure images

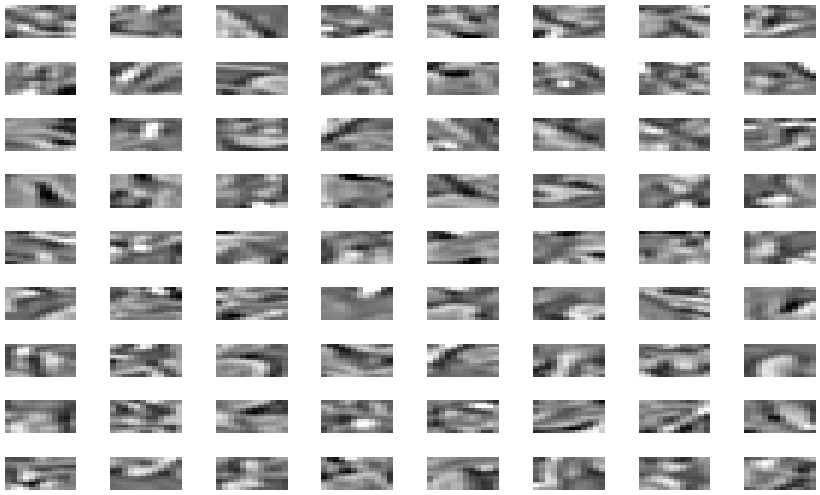


Figure 2.4: Random samples from natural scene images

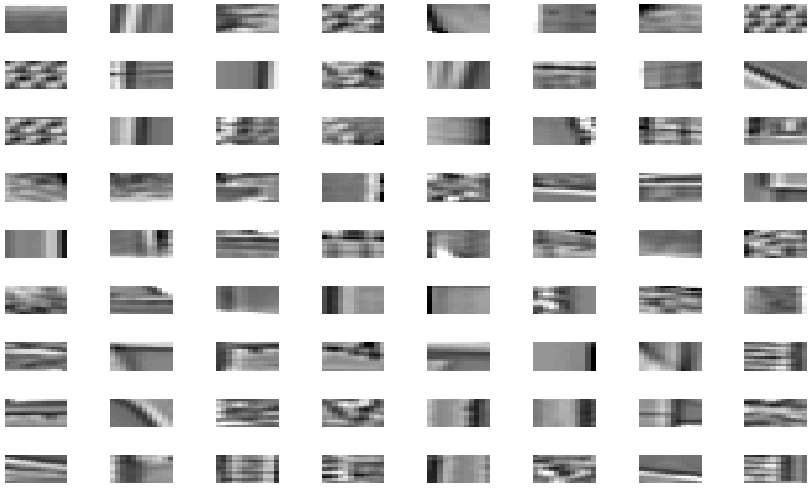


Figure 2.5: Random samples from man-made structure images

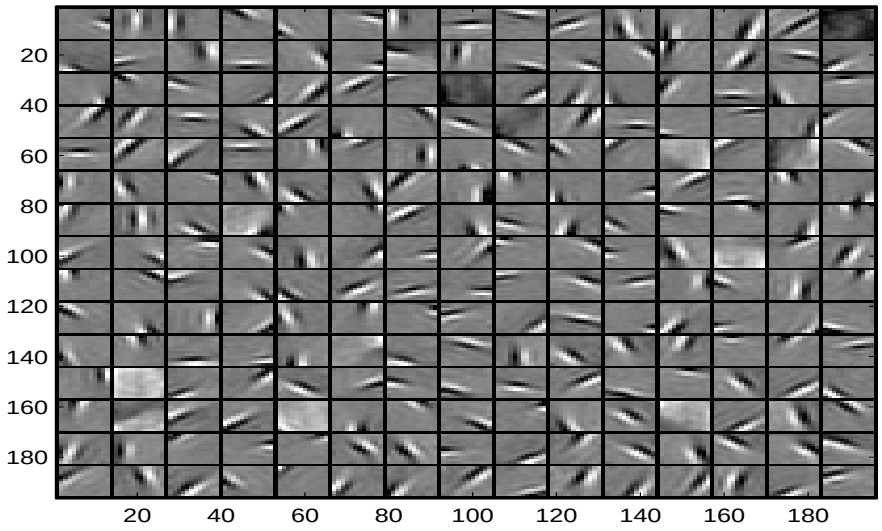


Figure 2.6: Basis functions for tree foliage

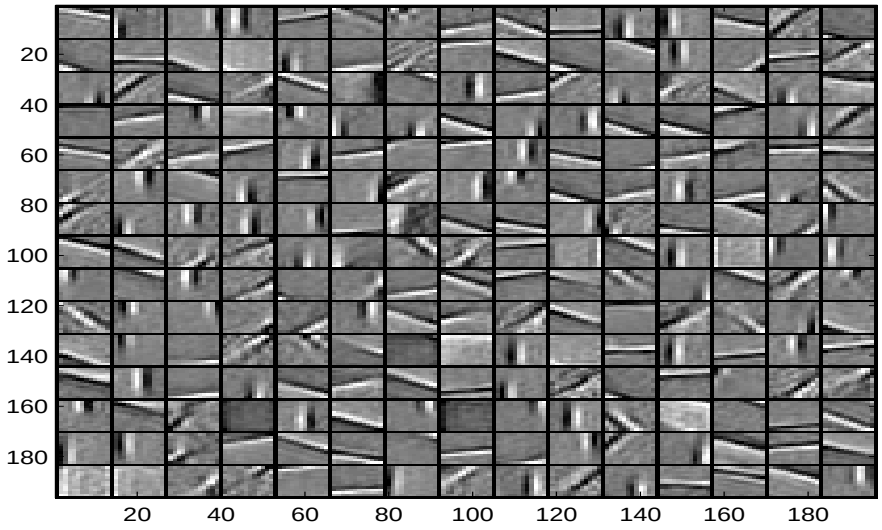


Figure 2.7: Basis functions for Man-made structures

2.2.3 Wavelet coefficients based basis function

The filters are learnt by adapting the wavelet coefficients to natural images as in [49]. The initial set of coefficients determine the scale and orientations of final outcome. The basis vectors are adapted by optimizing the cost function over thousands of randomly selected image patches in two steps: first, the coefficients are learnt and second, based on the new set of coefficients the basis functions are updated to reduce the error from the original image. This is repeated until a stable set of filters emerges. A typical set of basis functions and its spectra for 4 scales and 8 orientations are shown in Fig. 2.8. Please note that the filters are learnt from a natural stimuli database [21], which is completely different from the database used for the object recognition experiments.

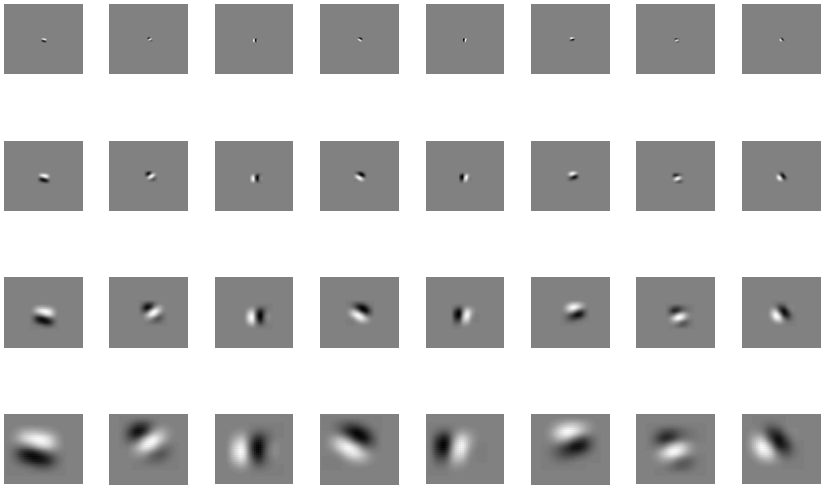
2.3 Measuring Sparseness

We measured the sparsensess of coefficients generated by random seed based coefficients. To measure sparseness Field [18] proposed using Kurtosis. There have other efforts [50, 56, 71] to measure the sparseness of the coefficients. Willmore & Tolhurst [71] introduced the concept of two different types of sparseness, explained as follows:

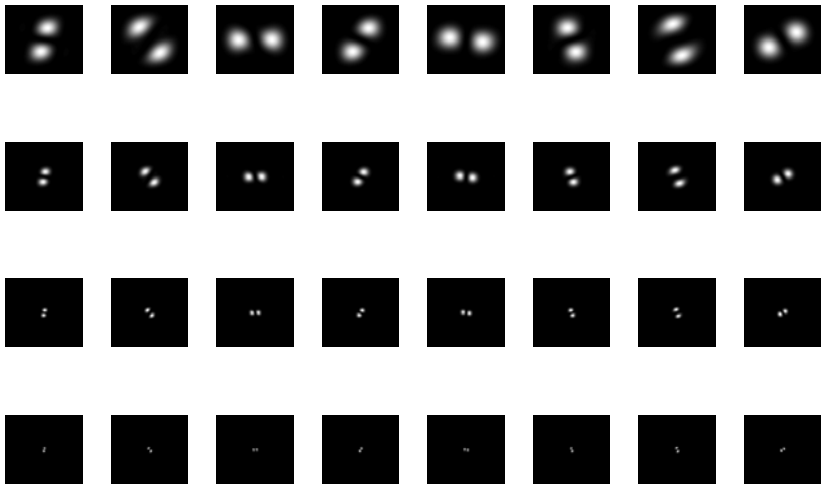
1. Population sparseness

2. Lifetime sparseness

In Population sparseness, for a given stimulus, only a small subset of coefficients respond; while in Lifetime sparseness for given set of stimuli a given coefficient rarely responds. Field meant sparseness in lifetime sense. As pointed out by Willmore & Tolhurst [71], one type doesn't imply the other. To measure both types of sparseness three methods were used and these produce similar results. These three methods are as follows:



(a) The learnt 8-orientation filters in spatial domain



(b) The learnt 8-orientation filters in frequency domain

Figure 2.8: Natural-stimuli adapted filters and their spectra produced using a modification of [47, 48].

2.3.1 Kurtosis

Kurtosis is a measure of "peakedness". A distribution with high sparseness will have greater value of the Kurtosis. Higher kurtosis means that distribution has sharper peak and flatter tail. The kurtosis is given by the following formula:

$$K = \left\{ \frac{1}{M} \sum_{i=1}^{M} \left[\frac{a_i - \bar{a}}{\sigma_a} \right]^4 \right\} - 3$$

Here, $\bar{a}$ is the average value of coefficients over M stimulations and σ_a is the standard deviation of the coefficients. One of the problems with kurtosis is that it is very sensitive to the outliers. Second, in real biological systems the firing rate is positive and hence the Kurtosis is one sided. Figure 2.9 (a) shows the increased kurtosis for sparse coding scheme.

2.3.2 Modified Treves-Rolls

Willmore & Tolhurst suggested a modification to account for only positive values. The only difference between this formulation and the original one proposed by Treves Rolls [56] is that it takes into account the absolute value of the coefficients. The modified formula is as follows:

$$S = 1 - \frac{\left[\sum_{i=1}^{M} |a_i| / M \right]^2}{\sum_{i=1}^{M} \left[a_i^2 / M \right]}$$

In this measure the value of S increases as the sparseness increases. Just like Kurtosis this measure can be used to characterize both lifetime as well as population sparseness. The behavior of our results based on this measure is shown in the figure 2.9(b). The trend in the curve confirms the fact that coefficients become more sparsified after learning.

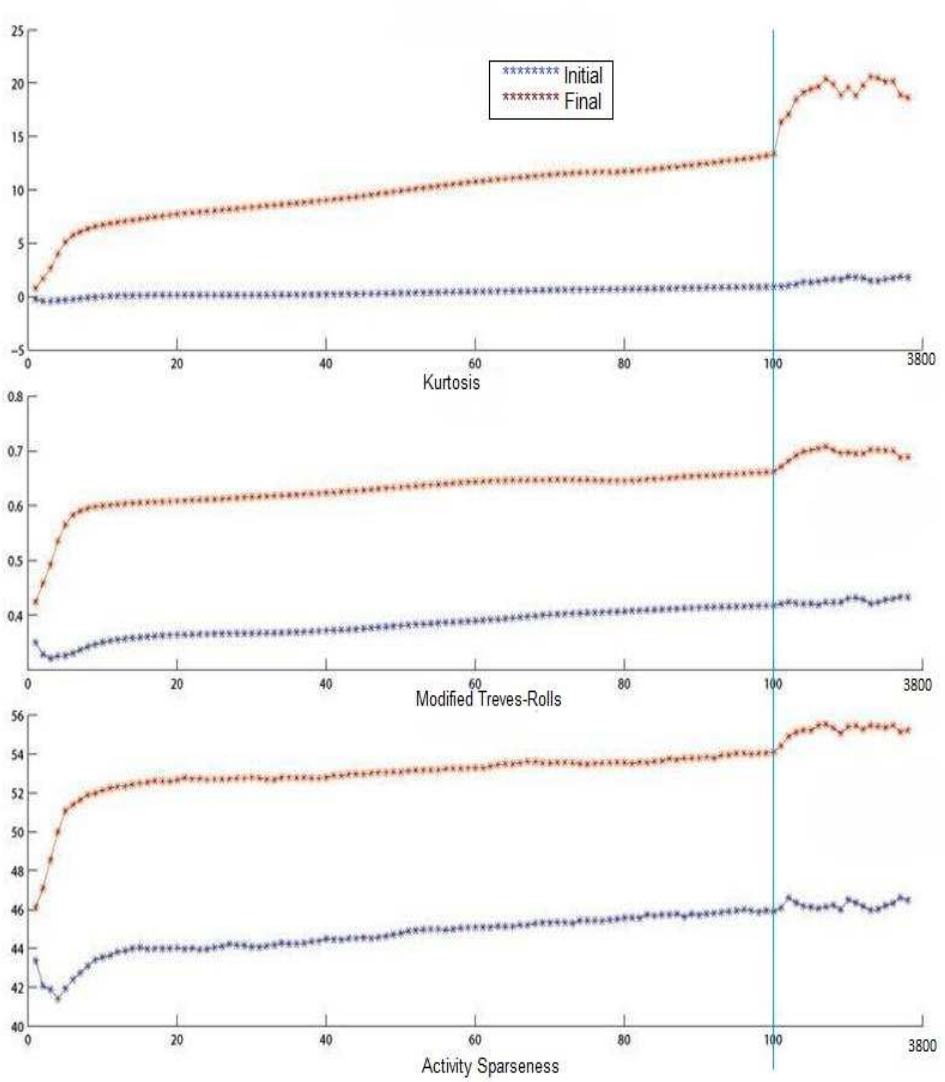


Figure 2.9: Lifetime sparseness measured over tree foliage dataset. “Initial” here means at the start of the iteration while “final” means the sparseness measure after stimuli was presented. The coefficients learnt were initialized randomly. Please, note that x-axis denotes the number of iterations while y-axis denotes the three sparseness measure mentioned in sec. 2.3.

2.3.3 Activity Sparseness

As the name suggest this measure is directly based on how many coefficients are active at any given point of time. To determine the activity at any given time, we have to define certain threshold value. The most obvious choice which we have used in this work is one standard deviation. The coefficients whose magnitude are within one standard deviation value of the mean are considered inactive. This measure basically counts how many coefficients (figure 2.9(c)) are at any given time "ON" or "OFF".

All the three measures described above can be used for both the population as well as lifetime sparseness. For measuring the population sparseness, first the response for a given stimulus is calculated and then it is averaged for all the stimuli present. For lifetime sparseness, the response of a particular coefficient (neuron) is calculated for a given set of stimuli and then it is averaged over all the coefficients. In this work, sparseness refers to lifetime sparseness.

It can be seen that qualitatively all three measures follow the same pattern. One problem as mentioned above with the kurtosis measure is that it is very sensitive to the outliers. Tolhurst and Willmore [71] carried out an extensive study to find the correlation between life time and population sparseness. It was shown that there is no correlation between them and one kind doesn't imply the other.

2.4 Advantages of Sparse Coding

Before moving on to next topic let's look at a few of the direct advantages of overcomplete representation and the resulting sparseness:

1. Simoncelli et. al [62] has shown that in critically sampled coding strategy a small perturbation in scale result in dramatic change in coefficients. They also argued that on the other hand overcomplete basis functions allow a smooth transition of coefficients.

2. J.H Van Hateren [20], Barlow [3] pointed out that having over complete set of receptive field (basis functions) translates into specificity in neural repose (coefficients) to certain features. This can be very attractive for pattern classification.

3. Overall the action potentials in the case of compact coding are higher than sparse coding [50]. This is in line with the fact that neurons should mostly stay inactive but when they respond they should have high magnitude [35, 71].

4. As pointed out by Olshausen & Field [50] advantage of overcomplete coding is that the data manifold becomes flattened. Hence, in overcomplete representation the probability of a particular receptive field (basis functions) responding to a signal becomes less. This specificity of receptive fields (to the signals) aids in the feature detection.

5. The hardware implementation [51, 59] of sparsely coded binary vectors (for associate memory) has been shown to produce minimum mean error rate of a pattern retrieval (for noisy data).

The sparseness coding shouldn't be confused with optimal compression schemes. With sparse coding the representation needn't be invertible. This means after learning is over coefficients have to be derived indirectly from the basis function. Sparse coding is appropriate when the given image has statistical redundancies and can be expressed in terms of few elementary features (e.g. lines, edges) [18].

Chapter 3

Feature Extraction Models

To tackle the multi-class object recognition problem as pointed out in chapter 1 we adopted a two step strategy: The feature extraction for the underlying dataset followed by a suitable classifier. In this chapter we will describe three classes of features which try to capture local, mid-level and global characteristics of the underlying dataset. The first set of features is based on a modification of the HMAX model, followed by modified gist and spatial pyramid based features, and then by mid-level features based on the Gestalt principle of continuity in visual perception.

3.1 Modified HMAX based C2 features

To date, mammalian visual systems have vastly superior performance for general vision related tasks. Risensheuber and Poggio [55] tried to incorporate lessons learnt from mammalian visual cortex into an HMAX model in order to build a computational model for multiclass object recognition. In the HMAX model, to maintain the trade-off between selectivity and invariance among different classes, features were combined by maximum pooling over space, orientations and scale sizes. It comprises a set of alternating simple- and complex-cell layers, and provides orderless feature extraction. One of the key steps in the HMAX model is to start with a set of predefined multi-orientation, multi-scale Gabor filters. We modify this step and include a set of filters adapted to the statistics of natural

stimuli. These filters and other modifications – along with the rest of HMAX model steps – are discussed below.

3.1.1 Pre-S1 Layer

In this step all the operations are carried out on grayscale images, hence if the incoming image is in color it is first converted to grayscale and then contrast normalized as in [42]. Each image is then reduced to 4 coarser resolution scales, where each scale is 1/1.25 times the previous one.

3.1.2 Simple Layer S1

In this layer the images from the previous step are convolved with natural-stimuli adapted filters. A typical set of these natural-stimuli adapted filters are shown in fig. 2.8. These natural-stimuli adapted filters are obtained as a result of optimizing Eq. 2.9 over thousands of image patches as described in sec. 2.2. The filters thus obtained are more attuned to the underlying statistics of the natural images and hence are more natural candidates for convolution than the Gabor filters as done in the original model [61]. During feature extraction the same size filter is convolved over progressively smaller images which results in lower computational cost than in [61].

This layer results in a multi-scale, multi-oriented representation of the given image. So, the total number of images after this layer for each input image is the product of the number of resolution levels and the number of orientations. In our case we have 4 resolution scales and 4 orientations resulting in 16 S1 layer images to represent each input image.

3.1.3 Complex Layer C1

The resulting images from the S1 layer are input to this layer. In the original model in this layer the information over successive resolution scales and within each location of a moving window of fixed size is combined. In our model due to empirical reasons we combine only the information within each location of a moving window of size 10×10. This is done by keeping the maximum pixel value over the window and thus keeping the most prominent of the edges. These windows overlap by half their size and result in some local invariance. So the number of resolution scales and orientations remains the same after this layer but the size of each image reduces. This layer also comes into significance when the model is learning the features for first time, as it is used to generate a dictionary of prototype S2 features in accordance with the bag-of-words models [10, 15]. The dictionary generation step is described in more detail in Sec. 3.1.6.

3.1.4 Simple Layer S2

In this layer responses from the C1 layer are used to compute the similarity from dictionary patches learnt after C1 layer in a sliding window fashion, similar to a convolution operation. To calculate the responses Euclidean distance is used as a similarity metric, though in practice other metrics can be also applied. For empirical reasons we calculate the response for 4 different sizes of patches in the dictionary: $4\times4, 8\times8, 12\times12$ and 16×16. We use the dense S2 features as all the information from different orientations is retained, as opposed to only prominent responses that are kept in [42]. So, *e.g.*, for each location in the C1 layer output and for a 16×16 S2 dictionary patch, a total of $16 \times 16 \times$ (number of orientations)

pixel computations are done and then added. Thus all the information from different orientations is combined and only the information for different scales is left. So after this layer the features incorporate rotation invariance to a certain extent.

3.1.5 Complex Layer C2

In this layer all the information from previous layers is combined. Similar to the original model, this is achieved by taking the maximum response over space and scale sizes of an image for each dictionary prototype patch. Hence, the number of C2 features after this layer is equal to the number of patches in the dictionary of prototype patches. Thus, each feature represents how similar the corresponding dictionary prototype patch is to the most similar patch in the input image, over all scales, orientations, and locations.

3.1.6 Dictionary of S2 prototype patches

A dictionary of S2 prototype patches is needed to calculate the response from the C1 layer. This dictionary of S2 patches is learnt by extracting patches from the C1 layer at random scales and locations. We follow a two-step method to generate a meaningful dictionary, as follows:

1. First, we generate a pool of candidate S2 prototype patches extracted form the C1 layer. These patches are randomly extracted from all locations over the image. Hence these patches might contain useful information *e.g.*, from portions of the object present in the image and its boundary, and also less useful information *e.g.*, from the background, sea or sky.

2. Second, to generate an efficient dictionary of patches from the pool we experimented with two different methods :

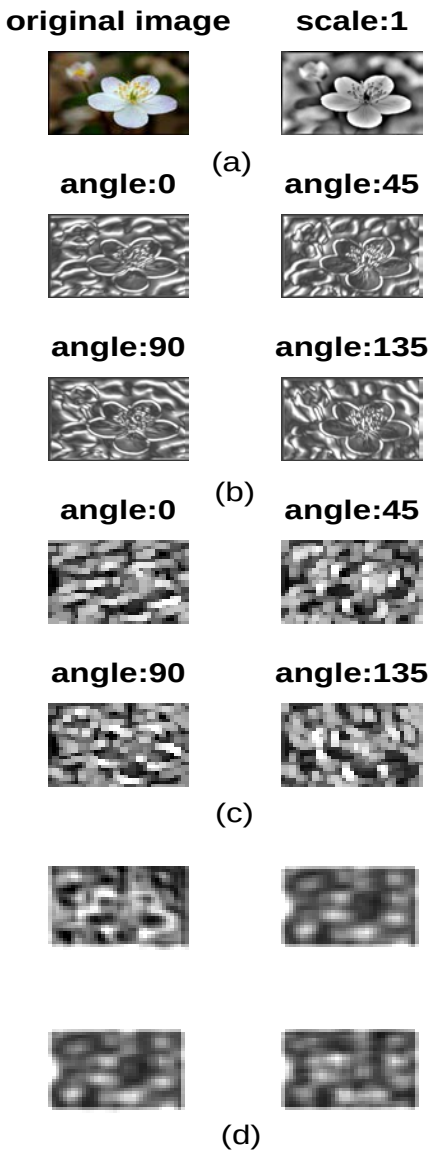


Figure 3.1: The response of different layers of modified HMAX model. In (a) the first image is the original image and the next image is the grayscale, resized, and locally contrast normalized version of it. In (b) the four orientation images of one resolution level of the S1 layer images are shown. In (c) the C1 features of the model for one resolution layer are shown. In (d) the responses of the S2 layer for 4 dictionary patches are shown, again for only one resolution layer.

(a) Mean patches resulting from *k*-means clustering.

(b) Patches that have largest L2-norm.

Based on experimental evidences we use *k*-means based dictionary generation.

There have also been attempts [60] to use a universal dictionary instead of the database dependent dictionary. Though these attempts lead to more generalized dictionary, they typically come at an expense of greater computational cost [60]

To illustrate the feature extraction process, a sample image from the Caltech-101 dataset is shown in Fig. 3.1 as it passes through various layers of this feature extraction method. The C2 features (not shown) will be a set of points which are the maximum S2 response corresponding to every dictionary patch. Thus the number of features is equal to the size of the dictionary.

3.2 Global feature

An object in a scene has strong correlation to the environment and thus the context in which it is present [45, 65]. It has been suggested that instead of taking the object-centric viewpoint, representing the object as a scene and thus taking the "holistic" viewpoint might provide important information. As an example, both water and sky are blue but it is easier to categorize them given their contexts *e.g.*, fish along with water and a bird in the sky provides more useful information. Context also attains importance in the case of poor imaging, low resolution or occlusion, the presence of noise, and texture representation [22, 65].

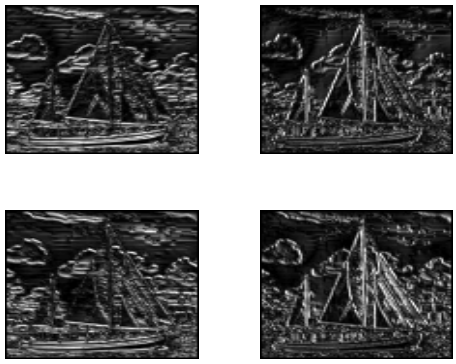


Figure 3.2: An example of typical modified gist features for 4 orientations and one scale. The original image is the first image shown in Fig. 3.4.

Modified HMAX based C2 features mostly represent intermediate-level features of an image using local patches, while continuity based Gestalt features (discussed below) capture mid-level features based on longer edges and contours. The local patch based features might have limitations in the presence of local noise and might not provide a useful representation for texture [27]. Hence, we include two sets of global features to take the context of images into account.

3.2.1 Modified Gist based global features

To address some of these problems Oliva and Torralba proposed using Gist based global features [45]. The Gist features capture contextual as well as appearance of the underlying object in a given image or scene. Following [26, 27] we include color channels along with the intensity channel for calculating the Gist features. We thus extend work of [45, 65] by incorporating color channels in addition to the intensity channel; the color-oppenency channels we use follow those of [69]:

$$RG = \frac{r - g}{\max(r, g, b)} \tag{3.1}$$

$$BY = \frac{b - \min(r, g)}{\max(r, g, b)} \tag{3.2}$$

in which *RG* accounts for red / green and *BY* accounts for blue / yellow double opponency; here *r, g,* and *b* are the red, green and blue channels of the input image. For the intensity channel we use ($I = 0.30\ r + 0.59\ g + 0.11\ b$) as in [8]. To capture Gist based features [65] use a steerable wavelet basis as a starting point. In our case we use the natural-stimuli adapted wavelet filters similar to those shown in Fig. 2.8.

The global characteristic of these features is obtained by averaging over a large spatial window. To further capture the shape information these spatial windows are tiled over the entire image in a manner similar to [8, 33]. The Gist features are calculated as follows:

$$m_c(x) = \sum_{x_a} I^k(x_a)\ w(x - x_a) \tag{3.3}$$

in which $I^k(x_a)$ is the output of the convolution with the natural-stimuli based filters, $m_c(x)$ is the Gist feature of either one of the opponenecy channels or the intensity channel, where c is the channel index, k is the current band (current resolution and orientation) being processed, and $w(x - x_a)$ is a uniformly averaging window outside which the energy is zero. We used 6 scales and 4 orientations of the original image to calculate the modified features. A total of 36 features per band were calculated resulting in $6 \times 4 \times 36 = 864$ features. We used these raw Gist features as our final features. In the original model the dimensionality of the raw Gist features was reduced and the decomposition coefficients were used to represent the contextual features. A typical set of modified Gist features is shown in Fig. 3.2.

3.2.2 Spatial pyramid based global features

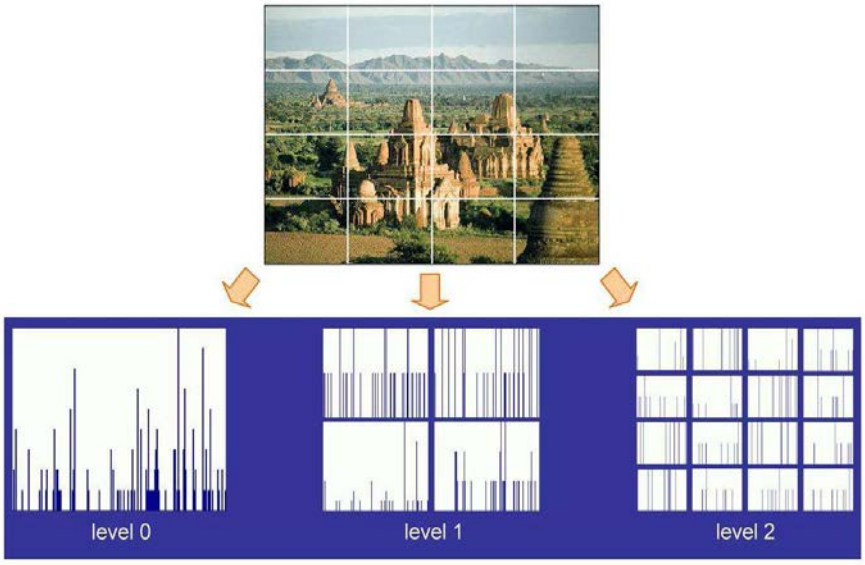


Figure 3.3: The spatial layout of the images for carrying out muiltscale feature extraction [33].

In the spatial pyramid model again a “holistic” view of the underlying images is taken for object categorization [33]. The spatial layout of the images is used for feature extraction by progressively dividing the given scene into finer sub-regions as shown in Fig. 3.3. A histogram from each sub-region is calculated and since no explicit account of relative geometric information is kept this method produces a set of non-invariant features. This technique doesn’t calculate explicit object models but uses a discriminative method based on these global features for object categorization.

The spatial pyramid matching scheme builds upon the multi-resolution pyramid match kernel of [19]. The features we used in this work utilize the “strong features” which are based on SIFT descriptors of 16×16 patches with a grid spacing of 8 pixels. In the spatial pyramid match all the features of the same type are quantized into M discrete types and

the assumption that features of the same type can match is made. The overall Kernel is the sum of the individual channel kernels:

$$K^L(X, Y) = \sum_{m=1}^{M} \kappa^L(X_m,\ Y_m) \tag{3.4}$$

where X_m and Y_m are the co-ordinates of the features. The $\kappa^L(X, Y)$ is given by the pyramid match kernel:

$$\kappa^L(X, Y) = \frac{1}{2^L}\mathcal{I}^{\,0} + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}}\mathcal{I}^{\,l} \tag{3.5}$$

where L denotes the number of spatial resolution levels and $\mathcal{I}$ denotes the histogram intersection function [19, 40] which is as follows:

$$\mathcal{I}(A, B) = \sum_{j=1}^{r} \min(A^{(j)}, B^{(j)}) \tag{3.6}$$

where A and B are histograms containing r bins and j^{th} bin in A is represented $A^{(j)}$.

3.3 Mid-level features based Visual perception

The modified HMAX based C2 model captures local patch-based intermediate features while the modified Gist based method captures global features. Bileschi and Wolf [5] suggested a set of mid-level features based on Gestalt principles of visual perception. This mid-level feature set is an ideal candidate for image representation along with the Gist and C2 features. In our work we use continuity based features, which are based on one of the Gestalt principles of visual perception as it captures the features based on longer continuous edges and contours by combining disconnected smaller edges.

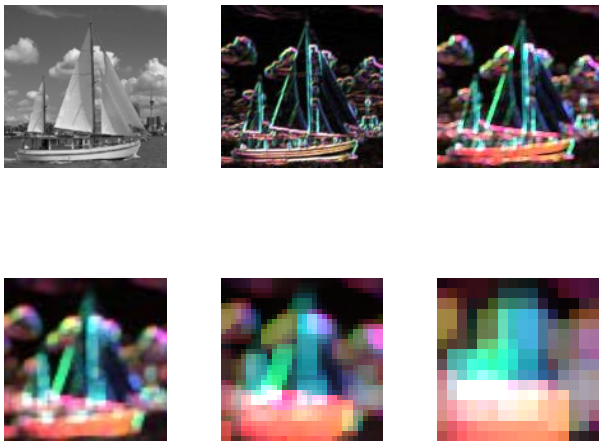


Figure 3.4: A typical set of mid-level features based on the Gestalt principle of continuity in visual perception [5]. The first image is the grayscale version of the original image. The next five images show the Gestalt features for different resolutions. The pseudocolor denotes the edge orientation and the color saturation denotes the relative strength of edges.

To extract the continuity based features, [5] suggested using a set of min-max based operations which is in contrast to the more traditional approach of combining the smaller candidate edges for extracting longer edges. In the original method the incoming image is first pre-processed by a set of equally spaced oriented filters (from 0° to 180°). These images are then processed by erosion and dilation morphological operators to carry out local minimization and local maximization. To generate a set of features for the next resolution a sub-sampling of 2×2 is performed. A typical set of resulting continuity-based Gestalt features are shown in Fig. 3.4.

Chapter 4

Feature Combination Methods

Combining the different feature sets for the common task of object recognition can be done in various ways. In practice there are two particular approaches which are mostly successful: first, the feature sets can be combined before they are fed to the classifier. In this approach successful methods include concatenating the individual feature sets [5] into one vector of higher dimension, and using mixture-of-Gaussians based methods.

In the second approach, each (individual) feature set is first used for classification and then the individual decisions or some measures of their confidence are used to make the final decision as done in multiple classifier systems (MCS) [14, 29, 31]. Multiple classifier systems include two predominant scenarios, namely distinct feature representations and similar feature representations. In the shared feature based methods different classifiers are assumed to estimate the same final functional and the final estimate is generally calculated by just taking the average of all the functionals. As an example one type classifier (*e.g.*, Neural network based) can be used multiple times, each with different parameters, and then in the end the final classification is based on taking the average over all the individual classifiers. It can also include techniques like logistic regression models [57], and boosting based methods [63]. In contrast in the distinct representation the overall result is based on all the individual but different functionals [14, 29].

Confidence measure based multiple classifier methods acquire importance especially if the distinct sets of features capture different characteristics of the underlying images. For the distinct features based combinations we follow the theoretical framework and the

extensive guidelines given in [29, 30]. One important point to be noted is that there are cases where distinction between two multiple classifier systems doesn't necessarily stay very sharp [28].

4.1 Classifier combination strategies

In the current work we first use each of the sets of distinct features (one local, one mid-level and two global) individually to calculate a confidence estimate based on posterior probability. These confidence measures can be combined using a trainable or non-trainable framework collectively called Class-Conscious Combiners [31]. The trainable techniques basically use weighted-average based combiners. For a more detailed treatment of these methods please refer to [30, 31].

The non-trainable framework here means that there are no more parameters needed for the combiner once the base classifiers are trained. The final classification result is the index corresponding to the $\max_i\{\mathcal{F}_i(\vec{x})\}$, where $\mathcal{F}_i(\vec{x})$ is generated by combining over all feature sets $j = 1, 2, ...L$, the individual confidence measures for each class $\omega_i (i = 1, 2, ...c)$. The different ways of combining the confidence measures, $\mathcal{F}_i(\vec{x})$, are as follows:

1. Product rule: This rule assumes that the different classifiers are independent (using posterior probability as the confidence measure [29]) of each other. Though in real-life scenario individual base classifiers are hardly independent of each other. For a given feature $\vec{x}$, the combination rule is as follows:

$$\mathcal{F}_i(\vec{x}) = \prod_{j=1}^{L} C_{i,j}(\vec{x}) \tag{4.1}$$

where $C_{i,j}(\vec{x})$ is the outcome of classifier j for class $\omega_i (i = 1, 2, ...c)$. The caution with the product rule is that the result can be noisy and also it may fail if the estimates are zero or very small [31]. The product rule can be also viewed as the product-of-experts method proposed in [23] and it was applied to the task of object recognition in [64].

2. Harmonic mean:

$$\mathcal{F}_i(\vec{x}) = \left(\frac{1}{L} \sum_{j=1}^{L} \frac{1}{C_{i,j}(\vec{x})} \right)^{-1} \tag{4.2}$$

3. Arithmetic mean:

$$\mathcal{F}_i(\vec{x}) = \frac{1}{L} \sum_{j=1}^{L} C_{i,j}(\vec{x}) \tag{4.3}$$

4. Maximum:

$$\mathcal{F}_i(\vec{x}) = \max_j \ C_{i,j}(\vec{x}) \tag{4.4}$$

Please, note that all the combination rules have been shown to be special cases of the product rule [14, 29] and that there is no guideline suggesting preferential treatment of one method over an other [30]. It has been pointed out in [31] p. 160 that the product based and arithmetic mean based methods are the most popular.

In this work we use posterior probability as a confidence measure. The posterior probability obtained in our case is based on Platts [7] formulation of Support Vector Machine (SVM) based decision boundaries. The reason for selecting SVM is that they have been shown to do well on large and sparse datasets as the final confidence measure lies on few prototypes or Support Vectors [7, 13, 24]. To carry out multiclass experiments we used a set of one vs. all based methods for SVM [24].

We conducted experiments based on both trainable and non-trainable combiners. Even the worst case non-trainable combiner comfortably outperforms the weighted average

based trainable classifiers. So, for final calculations we used non-trainable multiple classifier systems.

4.2 Single- vs. Multiple- Kernel Learning

To calculate probability based on individual feature sets we used support vector machines. For general classification tasks best results have been reported for linear or Gaussian kernels [24]. To choose between these kernels we make the decision based on the datasets by using multiple-kernel learning (MKL) [2, 67]. MKL uses a linear combination of a pre-specified set of kernels with coefficients that are learned from the data. As in [67] we used LIBSVM [24] for SVM solver. The discriminant function $f(x)$ is expressed as follows [7, 67]:

$$f(x) = \sum_{i=1}^{l} \alpha_i^* y_i K(x,\ x_i) + b^* \tag{4.5}$$

where $\{x,\ x_i\}_{i=1}^{l}$ are the l training examples which belong to $y_i \in \{+1, -1\}$, and $K(x,\ x_i)$ are the pre-computed positive definite kernels. α^*, b^* are parameters which are to be learnt from the examples. In the MKL framework the kernel is given by a linear combination of base kernels as follows:

$$K(x,\ x^{'}) = \sum_{k=1}^{M} d_k K_k(x,\ x^{'}) \tag{4.6}$$

where M denotes the total number of kernels. Here each K_k is a classical kernel. Learning both d_k and α_i in a single optimization is the core of MKL kernel learning. We follow the two-step optimization process as in [67] to learn these parameters.

In the current work the overall classification is based on combining the posterior probability of the individual classifiers, hence we use the probability estimate based classification to compare the performance of an MKL machine to a linear-kernel SVM. We conducted the experiment on the intensity component of Gist features.

The probability estimate for the multiclass problem is based on pairwise estimation [72]. So, to calculate the multi-class probability from the pairwise probability estimate of the linear combination of elementary kernels we use the second formulation of [72] which solved the following objective function:

$$\min_{\mathbf{p}} 2\mathbf{p}^T Q\mathbf{p} \equiv \min_{\mathbf{p}} \frac{1}{2}\mathbf{p}^T Q\mathbf{p} \tag{4.7}$$

where $\mathbf{p}$ is the probability estimate and Q is defined as follows:

$$Q_{ij} = \begin{cases} \sum_{s:s\neq i} r_{si}^2 & \text{if } i = j, \\ -r_{ji}r_{ij} & \text{if } i \neq j. \end{cases} \tag{4.8}$$

where r_{ij} are the pairwise probability estimates for the class i and class j. The advantage of this formulation over others is that it has been shown to be theoretically and empirically stable and to work better with large datasets [72].

We conducted 10 independent runs to decide between multiple kernel learning and a linear kernel for probability based classification. The result (percent correct) for MKL based classification was 37.07±0.7289 and for linear kernel was 38.72±1.27. As mentioned we used the intensity based gist features and the results for gist features were typical of other features used in this work. Even though the results were almost equal we chose linear kernel based probability estimation because it was slightly better as well as faster than the MKL based probability estimates.

Chapter 5

Datasets and Experiments

5.1 Dataset

To carry out object recognition experiments a suitable dataset is needed. The datasets should be representative of real world objects found in natural surroundings. At the same time it shouldn't be too difficult to carry out a controlled set of experiments for object recognition tasks using different computational models. The key aspects expected in a *well-balanced* dataset are: intra-class variability, pose, occlusion, viewpoint, background, scale, lighting etc. [16, 52].

To test the robustness and suitability of our feature extraction and classification model, in this work we used three different datasets: Caltech-101[15], Caltech-5[15], and Oxford 17-Flowers[44]. The brief description of these datasets along with their key characteristics are described below.

5.1.1 Caltech-101

The Caltech-101 dataset in total has 9194 images from 101 object categories and one background category. The number of images per class varies from a minimum of 31 to a maximum of 800, and most of the images are of size 300×300 pixels. The total number of 102 categories provides an ample opportunity to test any multiclass object recognition technique and it focuses on testing inter-class variability of the classification models [15, 53]. It has also been pointed out that this was created to test a system [52] capability for the

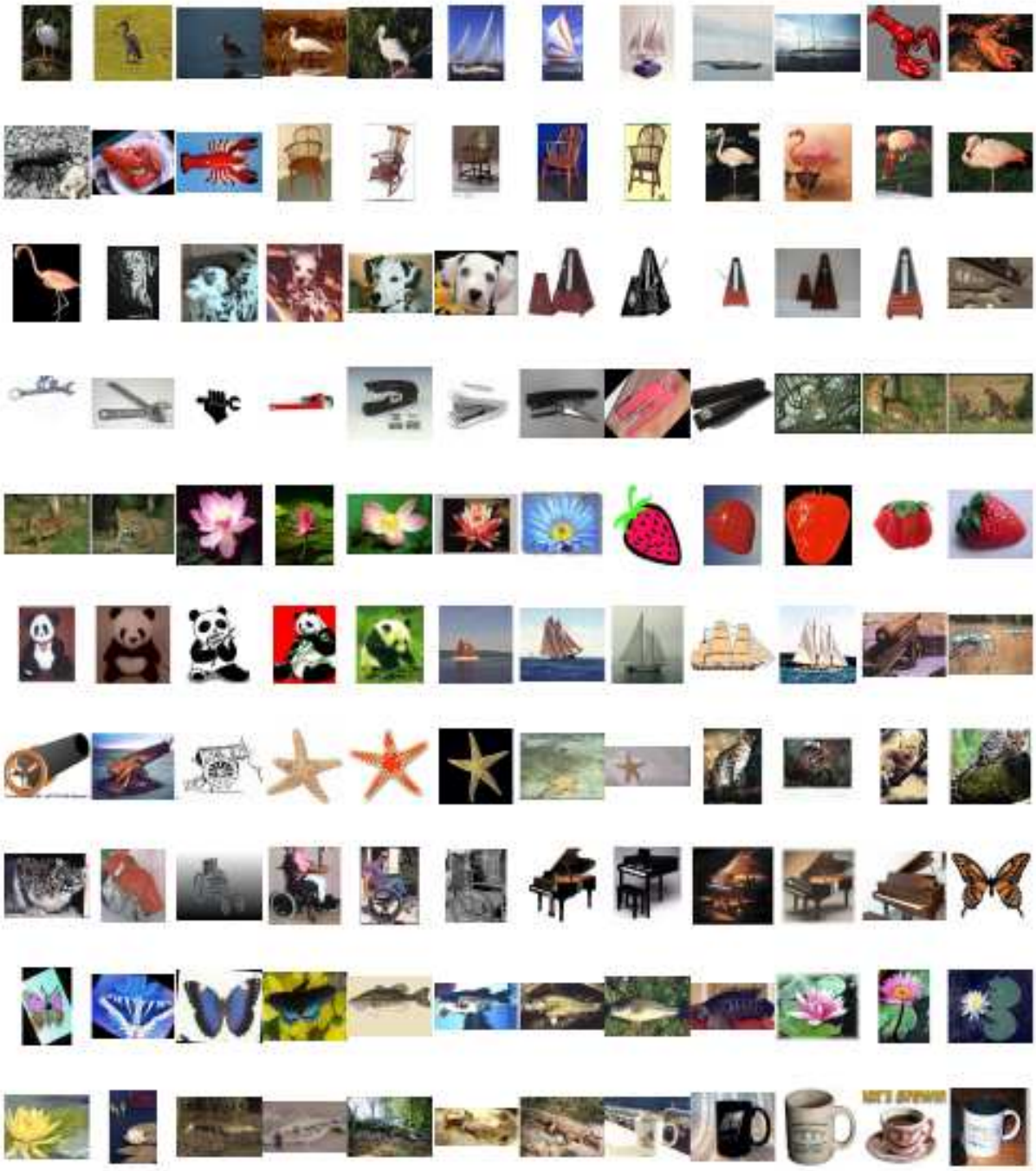


Figure 5.1: Five randomly selected images from each of 24 of the image categories (picked randomly) of Caltech-101. Overall there are a total of 102 categories in this dataset.

more difficult problem of object categorization assuming it is already position, scale and pose invariant [15, 52].

It has been pointed out that this dataset has some drawbacks as well [15, 52]: some of its images are well-centered and have very simple background; also the skewed distribution of images across the different classes poses the problem of fairly dividing the number of images per class for training and testing of the images. Also lack of position, scale and pose invariant testing ability of this dataset makes it less difficult and people can exploit this weakness [52, 53].

A typical set of randomly extracted images from different categories of Caltech-101 dataset is shown in Fig 5.2.

5.1.2 Caltech-5 dataset

This dataset is a subset of above-mentioned Caltech-101 dataset. The object categories in this dataset are: Leaves, airplanes, cars, motorbikes and faces. We used this dataset to calculate the ROC characteristics of our classification model. This was done along the lines to make sure not to rely only on classification accuracy of the model but to test it also for false positives. The detailed setup for conducting experiments along with some key results are mentioned in sec. 6.1.

5.1.3 Oxford Flowers dataset

The Oxford flowers dataset [44] comprises 17 different species and each species has 80 images. The dataset set has been compiled in such a manner that different kinds of features [44, 66] *e.g.*, color, shape etc., are needed to capture characteristics of different species. Reference [44] points out that the dataset has large intra-class variability and in some cases

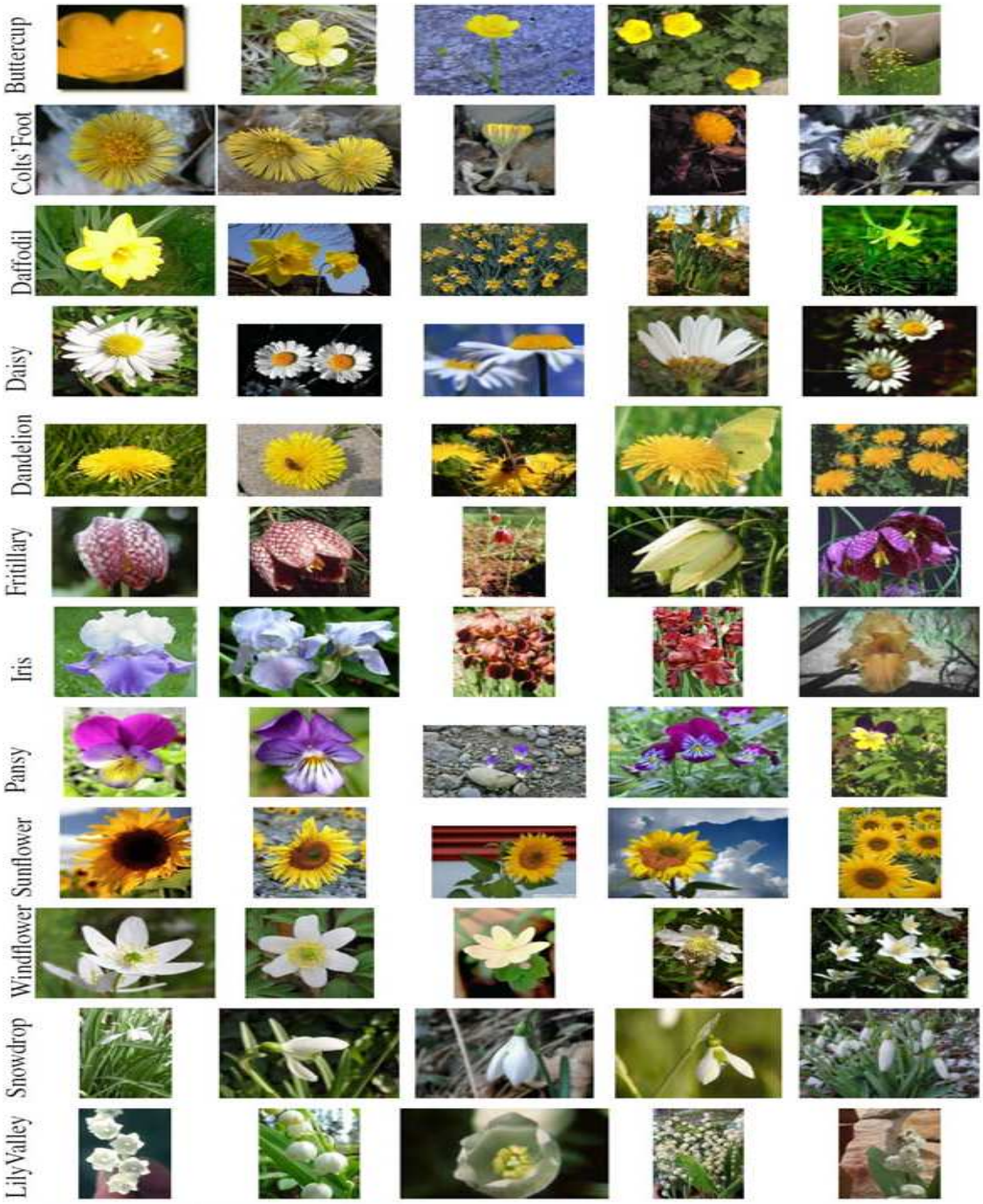


Figure 5.2: Few of the typical images in the Oxford flower dataset. Overall there are a total of 17 categories in this dataset.

smaller inter-class variation. Further the dataset has large viewpoint, scale and illumination variation. We chose this dataset to test our feature extraction models and classification method for these features as Caltech-101 is known [52, 53] not to possess some these key characteristics.

5.2 Experimental Setup

5.2.1 Caltech-101 dataset

For conducting experiments we used the method suggested in [15] and then followed [61]. For extracting modified HMAX features we used the following steps:

1. For training we used 15 to 30 images per class while for testing we used a maximum of 50 images per class. For classes in which less than 50 images were left, all the remaining images were used for testing.

2. To extract the dictionary of S2 patches a pool of C1 features were learnt from training images. Then a dictionary of size 2000 patches was selected using the *k*-means algorithm.

3. After the dictionary of patches was learnt, C2 features were calculated as a response to these patches.

Based on empirical results, 2000 modified HMAX features per input image are calculated during testing and training. These modified HMAX features were learnt from patches of sizes $4 \times 4, 8 \times 8, 12 \times 12$ and 16×16, and the corresponding number of features were 200, 400, 700 and 700, respectively. To extract color Gist features we used 6 resolutions and 4 orientations for each image. A total of 864 features were produced from a total of 24

channels each of which contributed 36 features. For the second set of global features (based on spatial pyramid matching) we used the default parameter values suggested in [33], *i.e.*, a dictionary of size 200 and the resolution level L = 2 was used, which resulted in a total of 4200 features per image. For the continuity based Gestalt features we used 4 resolutions and a set of orientations between 0 and 180 degrees. A total of 6120 features per image were used for training and testing of images.

5.2.2 Caltech-5 dataset

In this step we ran the experiments for testing the $object \mid \overline{object}$, *i.e.*, object presence vs. object absence, capability of the model. Towards this end we randomly selected 20 images for training and 50 images for testing (per category) from the object and the background categories.

5.2.3 Oxford Flower dataset

To conduct our experiments we use 40 training and 20 test images per class as in [44]. There are a couple of important differences between our experimental setup and [44, 68]. We do not include the validation set as used in [44, 68]. Also, we do not use segmentation of objects from the background as used in both [44, 68], which makes the classification task more difficult in our case. This is done to be consistent with the experimental setup for Caltech-101 dataset. To calculate the C2 features we follow the same setup as for Caltech-101 except 200 features per patch size are used resulting in 800 features per image. Both the Gestalt features and color Gist features are calculated in a similar manner as for Caltech-101 and also the feature vector is of the same size.

Chapter 6

Results

The results for Caltech-5, Caltech-101 and Oxford Flowers datasets are described below.

6.1 Caltech-5

In this step we ran the experiments for testing the $object \mid \overline{object}$, *i.e.*, object presence vs. object absence, capability of the model. Towards this end we randomly selected 20 images for training and 50 images for testing (per category) from the object and the background categories. For this experiment, we only used modified HMAX features and the feature vector of size 800 per image is used. The average result (mean and standard deviation) over 6 independent runs is shown in Table 6.1.

Object Category	Benchmark	Results
Leaves	84.0[70]	91.77 ± 3.20
Airplanes	90.2[17]	91.88 ± 2.09
Cars	88.5[17]	97.45 ± 1.89
Motorbikes	92.5[17]	92.62 ± 4.02
Faces	96.4[17]	89.63 ± 3.86

Table 6.1: Comparison (% correct classification) of our results for ($object|\overline{object}$) with the benchmark [17, 70].

It has been pointed out that just measuring the accuracy of results of an algorithm or a model can be misleading [12, 54]. Also, while evaluating the performance of a system it is

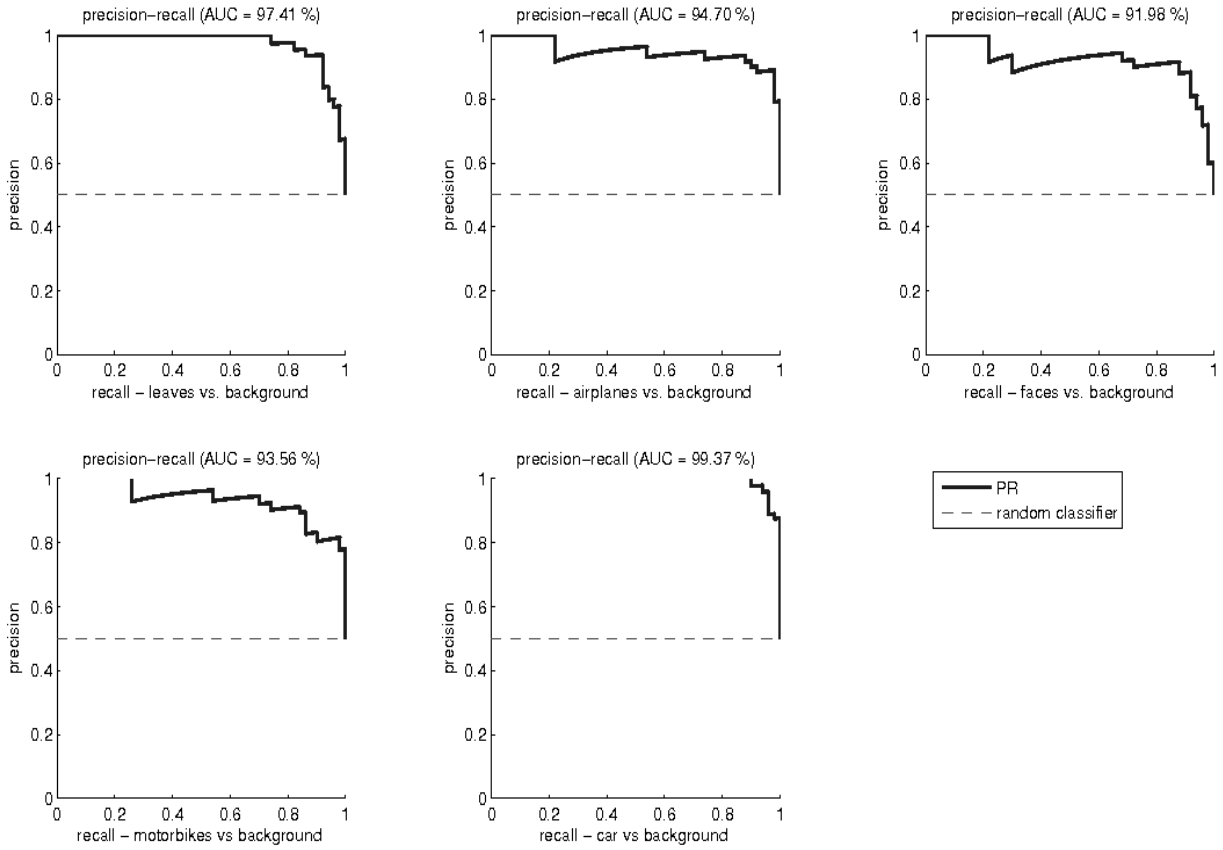


Figure 6.1: Precision-Recall curve for Caltech-5 dataset. AUC stands for area under the curve.

important to know how many objects are detected and how often there are false detections. Thus we evaluated the performance of the system using precision-recall curves. We use the following definition of precision and recall [1, 12]:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6.1}$$

$$\text{Recall} = \frac{TP}{nP} \tag{6.2}$$

Feature Sets	15/50	30/50
C2	33.04 ± 1.12	40.71 ± 1.28
Gest	37.37 ± 0.93	43.99 ± 0.87
GistC	40.02 ± 1.19	46.27 ± 1.88
SP	45.28 ± 0.45	55.32 ± 0.99

Table 6.2: Performance (percent correct) of individual classifiers. C2: modified HMAX C2 features; Gest: features based on Gestalt principle of continuity; GistC: modified color gist features; SP: Spatial pyramid based features. Each column heading gives the number of training and testing images. The results are based on an average of 10 independent runs for 15/50 case and 10 independent runs for 30/50 cases. Here 15 and 30 are the number of training images and 50 is the number of test images used.

in which TP is true positive, FP is false positive and nP is the total number of actual positives in the class. So, recall tells us about the correct probability of detecting positive test samples, while precision indicates the fraction of the correctly detected positive test samples [32]. It has been pointed out that in a precision-recall curve it is desirable to see how close the curve is to the upper-right corner [12]. One way to further measure performance is to calculate the area under the curve (AUC): the larger the area, the better the system performance. In Fig. 6.1 all the curves have more than 90% AUC, which points to the strong binary classification capability of the model. The precision-recall curves for all five categories are shown in Fig. 6.1.

6.2 Caltech-101

We started with choosing among the different fusion methods as mentioned in Sec. 4.1. Towards this end we took an empirical approach in which we conducted a set of experiments that used within each category 15 to 30 training images and 50 test images or less

Combiner	15/50	30/50
Arithmetic mean	56.77 ± 0.99	**64.10 ± 1.23**
Product	55.36 ± 0.70	63.31 ± 1.01
Harmonic mean	44.96 ± 0.75	52.10 ± 0.61
Maximum	49.54 ± 0.99	58.43 ± 1.00

Table 6.3: Comparison of classification performance based on combined feature sets; percent correct based on different fusion methods. All four feature sets were used. The results are based on an average of 10 independent runs for both 15/50 and 30/50 cases.

depending upon the number of test images left after training. The results based on individual feature sets are shown in Table 6.2 and the results of using each of the four combiners are shown in Table 6.3. It can be observed that the classification based on arithmetic mean outperforms all the other combination methods. Reference [30, 31] mentions that arithmetic mean and product based fusion methods give the best results with the arithmetic mean combiner being less noisy. Our results (Table 6.3) confirm that arithmetic mean and product mean based combiners perform better than the others. However, our results show that the product based combiner are stabler (as in low standard deviation) than arithmetic mean based combiner which is different from the observation made in [30, 31]. For the rest of the simulations in this paper we used the arithmetic mean based combiner for classification results.

To verify the contribution of individual feature sets we took a stepwise approach. If we are to follow [29] and if all four features sets capture different characteristics of the underlying images, addition of new feature set(s) should improve the classification results. We tested this idea for both 15 training and 50 test images as well as 30 training and 50 test images. The results are shown in Fig. 6.2 where first four bars of each plot show the contribution by individual feature sets. The following bars progressively show the addition

Feature Sets	C2	Gest	GistC	SP	All
Ant	5.35 ± 4.58	6.58 ± 4.45	6.58 ± 3.09	2.88 ± 4.05	13.99 ± 8.24
Cougar Body	0.69 ± 1.38	8.33 ± 5.85	7.99 ± 4.45	18.40 ± 6.53	16.32 ± 6.21
Wild cat	5.26 ± 6.45	4.68 ± 4.12	8.19 ± 7.02	15.79 ± 6.96	19.98 ± 10.78
Scorpion	8.00 ± 3.87	12.67 ± 8.19	12.44 ± 7.47	9.78 ± 4.94	20.44 ± 6.69
Beaver	3.58 ± 2.99	13.26 ± 0.99	11.11 ± 6.06	7.17 ± 6.61	16.85 ± 5.99

Table 6.4: Comparison of classification results (percent correct) for 5 most difficult classes over individual feature sets and their overall combination. The results are based on an average of 10 independent runs for 15/50.

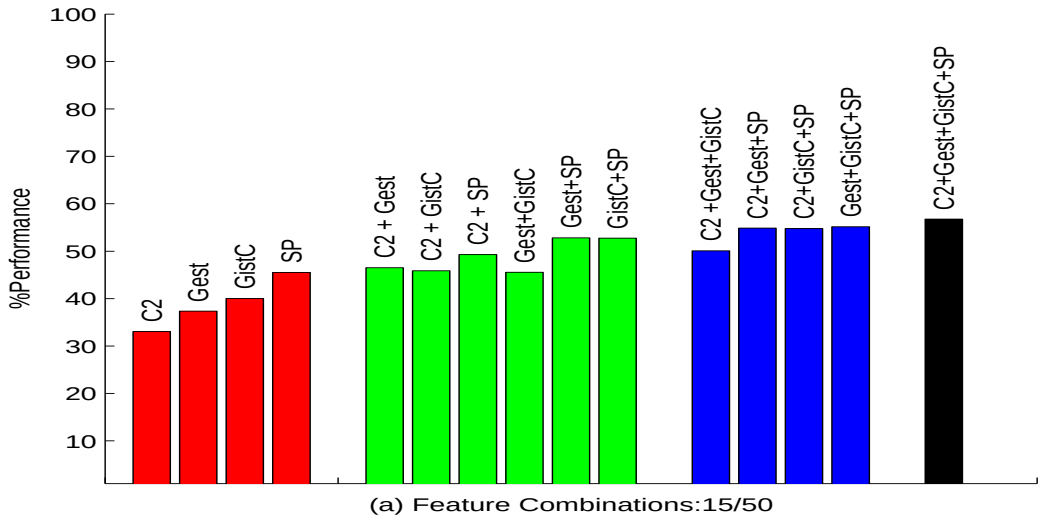
Feature Sets	C2	Gest	GistC	SP	All
Faces	50.44 ± 10.33	64.44 ± 9.99	46.66 ± 5.90	89.56 ± 7.54	97.23 ± 2.00
Motorbikes	71.23 ± 13.82	78.00 ± 9.43	89.33 ± 3.61	77.33 ± 10.49	91.56 ± 3.97
Accordion	78.06 ± 5.97	73.89 ± 7.92	66.94 ± 7.68	90.83 ± 6.37	93.61 ± 3.97
Trilobite	70.67 ± 7.94	73.33 ± 7.81	79.11 ± 5.49	93.56 ± 5.98	93.11 ± 4.81
Pagoda	82.29 ± 6.63	72.57 ± 9.86	86.46 ± 4.69	89.24 ± 9.26	94.45 ± 2.61

Table 6.5: Comparison of classification results (percent correct) for 5 highest performing classes over individual feature sets and their overall combination. The results are based on an average of 10 independent runs for 15/50.

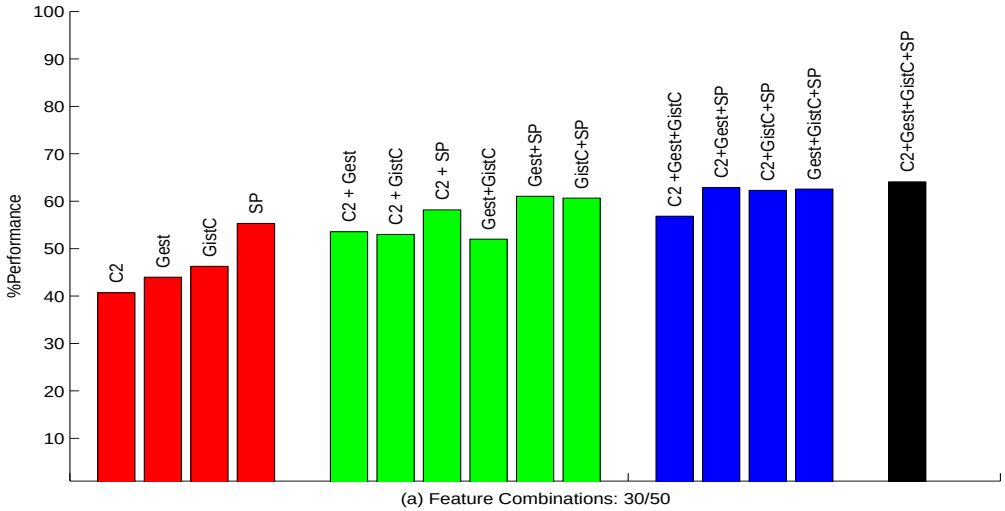
of new feature sets and the resulting improvement in the overall classification. The final bar of each plot, which combines all the feature sets, produces the best classification result, thus justifying bringing all the feature sets together for overall classification. All the features we train use a linear-kernel based support vector machine classifier for each feature set. To evaluate the overall result, the performance metric in Eq. 6.3 was calculated by averaging over all the individual class results, as per the guideline in [15].

$$Overall\ Accuracy = \frac{accClass_1 + accClass_2 + \ldots + accClass_N}{N} \tag{6.3}$$

where $accClass_i$ means accuracy for the i^{th} class and N is the total number of classes. Here $accClass_i$ means fraction of test data correctly classified for i^{th} class.



(a) The 15/50 results



(b) The 30/50 results

Figure 6.2: Comparison of different combinations of four feature sets, using the arithmetic mean combiner. C2: modified HMAX C2 features; GistC: modified color gist features; Gest: features based on Gestalt principle of continuity; SP: Spatial pyramid based features. The results are based on an average of 10 independent runs for both 15/50 and 30/50 cases.

To further investigate the effect of various feature sets across different classes we looked into five “easiest” and five “hardest” classes. These object classes were selected if they were 6 or more times (out of 10 runs) either in the worst or best classified object category. For the best performing class we didn’t take “Faces Easy ” and “car side” classes into consideration. These two object classes were easiest, as for every individual feature set 100% classification accuracy was achieved. Also, “Background” category was not considered for the hardest class as it is mostly an assortment of disparate groups of objects rather than one comprehensive object class.

Table 6.4 shows the interaction of different feature sets for the hardest five classes over 15 training and 50 test images. In most cases adding extra feature sets improves the classification result, in some cases markedly, but in one case the performance degrades. That is for “Cougar Body” object class the final performance actually deteriorates (relative to the SP result) after all the individual features are brought together. The impact of adding all the feature sets is more consistent in Table 6.5. In this table the overall result almost always improves after combining the individual feature sets. These results further advocate the importance of bringing different feature sets together for multiclass object classification.

6.2.1 Comparison of Results: Caltech-101

It can be observed that every feature type added improved the overall classification result, although the incremental gain tends to vary. The best results are clearly obtained when all four types of features are brought together. Our best result based on 30 training and 50 test images is 64.10 ± 1.23%.

To compare our results with some of the other recent results the immediate problem is to try to define a level playing field. Since our model combines four diverse sets of features

based on biological vision, human perception and spatial pyramid of images features, it will make sense to compare it with the Bileschi and Wolf [5], Serre *et al.* [60], Mutch and Lowe [43], and Sevetlana Lazebnik [33]. The result reported in our paper for 15 training and 50 test images is higher than that of [5, 43, 60]. We achieve a classification accuracy of 56.77 ± 0.99% where as in [5] it is 48.26 ± 0.91% and in case of [61] it is 44 ± 1.14%, in case of [60] it is 55.0 ± 0.90% and in [42] the result is 51% for 15 training images. For 30 training images our result of 64.10 ± 1.23% is higher than 56% as reported in [43]. Results for 30 training and 50 test images were not reported in [5, 60]. Our model also has the advantage that we mostly use a smaller dictionary size and fewer resolution levels and orientations than in these comparison works.

Our classification result (30 training and 50 test images) for the SP feature set only is 55.39 ± 0.99% which is based on the default parameters mentioned in [33, 34]. We were unable to duplicate their reported percent correct result of 64 ± 0.8% based on SP feature set. In all our experiments, the result based on a combination of all the other feature sets with the SP feature set significantly improves our classification accuracy. All the individual feature sets were combined using the posterior probabilities as described herein.

6.3 Oxford Flowers

Similar to the previous dataset we first conducted the experiments to choose among the four combiners. We used 40 training and 20 test images as in [44]. The classification results based on individual feature sets are shown in Table 6.6. It can be again observed from Table 6.7 that the arithmetic mean combiner outperforms the other fusion methods, although it also has the highest standard deviation. The best classification accuracy based

Feature Sets	40/20
C2	43.65 ± 2.98
Gest	34.12 ± 1.78
GistC	46.18 ± 2.41
SP	56.59 ± 2.67

Table 6.6: Performance of individual classifiers for Oxford Flowers dataset. The column heading gives the number of training and testing images.The results are based on an average of 5 independent runs.

Combiner	40/20
Arithmetic mean	**76.53 ± 1.46**
Product	75.94 ± 0.57
Harmonic mean	47.65 ± 2.92
Maximum	65.70 ± 0.87

Table 6.7: Comparison of classification performance for Oxford Flowers dataset based on combined feature sets; percent correct based on different fusion methods. All four feature sets were used. The results are based on an average of 5 independent runs.

on the arithmetic mean fusion method is 76.53 ± 1.46%. These results are based on 5 independent runs.

To verify the impact of progressively combining different feature sets we use the arithmetic mean combiner. The stepwise additions of different feature sets should show that the different features progressively contribute to classification accuracy. The results are shown in Fig. 6.3, where again the first four bars show the contribution of individual feature sets. The following bars progressively show the addition of new feature sets and thus the resulting improvement in the overall classification. The final bar of the plot, which combines all

the feature sets, shows the best classification result, thus justifying bringing all the feature sets together for overall classification.

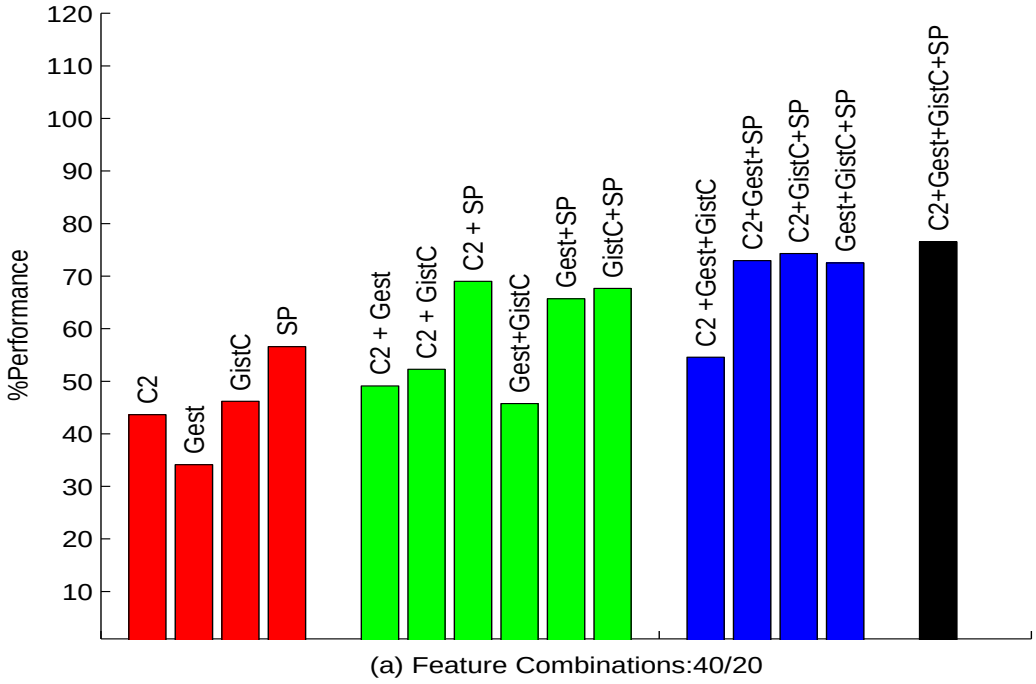


Figure 6.3: Comparison of different combinations of four feature sets, using the arithmetic mean combiner. The results are based on an average of 5 independent runs.

6.3.1 Comparison of Results: Oxford Flowers

In this dataset also it can be observed from Fig. 6.3 that the overall result is best when all four feature sets are used for the classification, although the incremental gain due to different feature sets varies. As mentioned earlier the best result for us is 76.53 ± 1.46 (Table 6.7). For this result we used 40 training and 20 test images as in [44, 68]. Having said that there are two very crucial differences. First, we don't use a validation set to adjust the parameters involved; second, unlike [44, 68] we don't use segmentation of objects from

background because our main aim is to classify the given data in the presence of various kinds of clutter and background. The best result obtained in [44] is 80.49 ± 1.97 and in [68] is 88.33 ± 0.3. We want to reiterate that in comparing our results with others there is a clear lack of level playing field and we are trying to solve a more difficult problem.

Chapter 7

Conclusion

In this work we revisit the multiclass object classification problem. We combine local features extracted from an object-centric viewpoint, mid-level features based on visual perception, and features extracted from a holistic viewpoint, to address the object classification problem. We demonstrate that appropriate feature sets extracted using different methods can be combined to provide better results on multiclass object recognition than with features extracted using any one (or any sub-combination) of the methods.

To capture the local features we modified the original HMAX model using L1-norm based natural-stimuli adapted filters to further the biological plausibility of the model. For the mid-level features we used visual perception based continuity features. These features capture longer edges and contours joining the smaller edges of the underlying scene. The context based global features were computed using a) Gist and b) spatial-pyramid based histogram-of-edges models. While all these features capture different characteristics of the underlying scenes, in general the overall improvement will be dependent on factors like complementarity of the different feature sets being combined along with the fusion method used and the problem at hand.

Different techniques for combining the feature sets were also examined. We took a slightly different approach than most of the methods in which some form of weighted average of the different features is taken before the classification is carried out. In this work we used discriminative-model based posterior probability as the base confidence measure. We examined two different ways of calculating the posterior probabilities: a) based on

linear combinations of Gaussian and Linear Kernels, and b) based on only linear kernels. For the given set of features both the techniques were shown to produce similar confidence measures for classification tasks. Among the different non-trainable fusion methods, we found that an arithmetic mean fusion method outperformed product, harmonic mean and maximum fusion methods.

7.1 Future Work

This work can be further extended in the following directions.

1. Bayesian approach to overcomplete representation: To calculate overcomplete representation in this work we followed maximum likelihood estimation as suggested in [48]. It can be further extended by using the Bayesian approach for estimating the prior density function.

2. Adaptive weights: In this work we used equal weights for all the confidence measures before combining them. In future work it would be informative to combine the confidence measures more adaptively *e.g.,* as in the committee-of-experts method. It would also be interesting to compare combining feature sets after classification (as in our work) with combining feature sets before classification.

3. Extending to color based features In this work we only utilized the color features (based on color opponency) for global Gist features. In future work extending the color features to other global and mid-level feature sets might also lend additional insight.

4. Shape, color or appearance: It will be interesting to explore the relative contribution of features based on appearance, shape or color on the task of object classification.

Fergus *et. al* [16] has explored it from a generative model perspective via mapping shape, color and appearance etc., to a Gaussian probability density. In [8] it is done using sweeping the relative effect on a grid of points. It will be interesting to explore this viewpoint in our framework.

5. Other applications: It would be interesting to compare the gain achieved by combining disparate feature sets using posterior probability in other machine learning problems.

References

[1] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[2] Francis R. Bach and Gert R. G. Lanckriet. Multiple kernel learning, conic duality, and the smo algorithm. In *In Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

[3] H. B. Barlow and P. Foldiak. Adaptation and decorrelation in the cortex. In G. J. Mitchinson C. Miall, R. M. Durbin, editor, *The computing Neuron*, pages 54–72. Addison-Wesley, 1989.

[4] A. J. Bell and T. J. Sejnowski. The independent components of natural scenes are edge filters. *Vis. Res.*, 37:3327–3338, Dec 1997.

[5] S. Bileschi and L. Wolf. Image representations beyond histograms of orientations: The role of gestalt descriptors. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[6] S.M. Bileschi. Object detection at multiple scales improves accuracy. In *Proceedings of ICPR*, 2008.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *CIVR '07*, pages 401–408, 2007.

[9] David D. Cox and James J. Dicarlo. Untangling invariant object recognition, 2007.

[10] G. Csurka, C. R. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. European Conf. Computer Vision*, 2004.

[11] J. G. Daugman. Entropy reduction and decorrelation in visual coding by oriented neural receptive fields. *IEEE Transactions on Biomedical Engineering*, 36:107–114, 1989.

[12] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 233–240, New York, NY, USA, 2006. ACM.

[13] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[14] R.P.W. Duin. The combining classifier: To train or not to train? In *Proceedings 16th International Conference on Pattern Recognition (ICPR)*, pages 765–770, 2002.

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

[16] R. Fergus. *Visual Object Category Recognition*. PhD thesis, University of Oxford, 2005.

[17] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of CVPR*, volume 2, pages 264–271, June 2003.

[18] D. J. Field. What is the goal of sensory coding? *Neural computation*, 6:559–601, 1994.

[19] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, Oct 2005.

[20] J. H. Haternen. A theory of maximizing sensory information. *Biological Cybernatics*, 68, 1992.

[21] J. H. Haternen and A. van der Schaaf. Independent component filters of natural images compared with simple cell in primary visual cortex. *Proc. R. soc. London*, 265, B 1998.

[22] X. He, R.S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR04*, pages II: 695–702, 2004.

[23] Geoffrey Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:2002, 2000.

[24] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, Mar 2002.

[25] D. H. Hubel and T. N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195:215–244, 1968.

[26] L.M. Hurvich and D. Jameson. An opponent-process theory of color vision. *Psychological Review*, 63:384404, 1957.

[27] 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*, 20(11):1254–1259, Nov 1998.

[28] Josef Kittler. A framework for classifier fusion: Is it still needed? In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 45–56, 2000.

[29] Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998.

[30] L.I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on PAMI*, 24(2):169–191, Feb 2002.

[31] L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience Publication, 2004.

[32] Thomas C. W. Landgrebe, Pavel Paclik, and Robert P. W.Duin. Precision-recall operating characteristic (p-roc) curves in imprecise environments. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 123–127, Washington, DC, USA, 2006. IEEE Computer Society.

[33] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of CVPR*, volume II, pages 2169–2178, 2006.

[34] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Spatial pyramid matching. In B. Schiele S. Dickinson, A. Leonardis and M. Tarr, editors, *Object Categorization: Computer and Human Vision Perspectives*, chapter 21, pages 401–415. Cambridge University Press, 2009.

[35] P. Lennie. Cost of cortical computation. *Current Opinion in Neurobiology*, 13:493–497, 2003.

[36] M. S. Lewicki and B. A. Olshausen. A probabilistic framework for the adaptation and comparison of image codes. *Journal of the Optical Society of America A*, 16(7):1587–1601, 1999.

[37] M. S. Lewicki and B. A. Olshausen. Probabilistic framework for the adaptation and comparison of image codes. *Optical Society of America*, 16(7), 1999.

[38] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[39] P. Mishra and B. K. Jenkins. Multiclass object recognition with sparse, localized features. In *2010 IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 2010.

[40] D.H. Ballard M.J. Swain. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[41] K. Murphy, A. Torralba, D. Eaton, and W. T. Freeman. Object detection and localization using local and global features. In *Towards Category-Level Object Recognition. Springer Lecture Notes in Computer Science (unrefreed)*, 2006.

[42] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *Proceedings of CVPR*, pages 11–18, June 2006.

[43] Jim Mutch and David G. Lowe. Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision (IJCV)*, 80(1):45–57, October 2008.

[44] M-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.

[45] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[46] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[47] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37:3311–3325, Dec 1997.

[48] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[49] B.A. Olshausen, P. Sallee, and M. S. Lewicki. Learning sparse image codes using a wavelet pyramid architecture. In *Advances in Neural Information Processing Systems*, volume 13, pages 887–893, 2001.

[50] Bruno A Oshausen and David J Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14:481–487, 2004.

[51] G. Palm and F. T. Sommer. Information capacity in recurrent mcculloch-pitts networks with sparsely coded memory states. *Network*, 3:177–186, 1992.

[52] Nicolas Pinto, David D. Cox, and James J DiCarlo. Why is real-world visual object recognition hard? *PLoS Comput Biol*, 4, 01 2008.

[53] Jean Ponce, T. L. Berg, M. Everingham, D. Forsyth, M. Hebert, Svetlana Lazebnik, Marcin Marszałek, Cordelia Schmid, C. Russell, A. Torralba, C. Williams, Jianguo Zhang, and Andrew Zisserman. Dataset issues in object recognition. In *Towards Category-Level Object Recognition*, pages 29–48. Springer, 2006.

[54] F. J. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–453, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[55] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, Nov 1999.

[56] E. T. Rolls and M. J Tovee. Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *The Journal of Neurophysiology*, 73:713–26, 1995.

[57] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, July 1990.

[58] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l1-regularization: A comparative study and 2 new approaches. In *Proc. of Euro. Conf. on Machine Learning*, 2007.

[59] F. Schwenker, F. T. Sommer, and G. Palm. Iterative retrieval of sparsely coded associative memory patterns. *Neural Networks*, 9:445–455, 1996.

[60] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, Mar 2007.

[61] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Proceedings of CVPR*, June 2005.

[62] E. P. Simoncelli, W. T. Freeman E. H. Adelson, and D. J. Heeger. Shiftable multiscale transform. *IEEE Transactions on Information Theory*, 3:587–607, 1992.

[63] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[64] A Torralba. Contextual priming for object recognition. *International Journal of Computer Vision*, 53(2):169–191, Jul 2003.

[65] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *IEEE Intl. Conference on Computer Vision (ICCV)*, Oct 2003.

[66] M. Varma and B. R. Babu. More generality in efficient multiple kernel learning. In *Proceedings of the International Conference on Machine Learning*, pages 1065–1072, June 2009.

[67] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil*, October 2007.

[68] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision*, September 2009.

[69] Dirk Walther. *Interactions of visual attention and object recognition : computational modeling, algorithms, and psychophysics*. PhD thesis, California Institute of Technology, Pasadena, California, 2996.

[70] M. Weber, W. Welling, and P. Perona. Unsupervised learning of models of recognition. In *Proc. European Conf. Computer Vision*, volume 2, pages 1001–1108, 2000.

[71] B. Willmore and D. J. Tolhurst. Characterizing the sparseness of neural codes. *Network: Computation in Neural Sysytems*, 12:255–270, 2001.

[72] T.-F Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.*, 5:975–1005, 2004.