Image and Video Classification and Image Similarity Measurement by Learning Sparse Representation

by

Tanaya Guha

MASc, University of Windsor, ON, Canada, 2008

BE, Bengal Engineering and Science University, Shibpore, India, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Electrical and Computer Engineering)

The University Of British Columbia

(Vancouver)

September 2013

Abstract

Sparse representation of signals has recently emerged as a major research area. It is well-known that many natural signals can be sparsely represented using a properly chosen dictionary (e.g. formed of wavelets bases). A dictionary could be complete or overcomplete depending on whether the number of bases it contains is the same or greater than the dimensionality of the given signal. Traditionally, the use of predefined dictionaries has been prevalent in sparse analysis. However, a more generalized approach is to *learn* the dictionary from the signal itself. Learnt dictionaries are known to outperform predefined dictionaries in several applications.

This thesis explores the application of sparse representations of signals obtained by learning overcomplete dictionaries for three applications: 1) classification of images and videos, 2) measurement of similarity between two images, and 3) assessment of perceptual quality of an image.

This thesis first capitalizes on the natural discriminative ability of sparse representations to develop efficient classification algorithms. The proposed algorithms are employed in image-based face recognition and video-based human action recognition. They are shown to perform better than the state-of-the-art.

The thesis then studies how to obtain a good measure of similarity between two images. Despite the long history of image similarity evaluation, open issues still exist. These include the need of developing generic similarity measures that do not assume any prior knowledge of the task at hand or the data type. This thesis develops a generic image similarity measure based on learning sparse representations. Successful application of the proposed measure to clustering, retrieval and classification of different types of images is demonstrated.

The thesis then examines a highly promising approach to assess the perceptual

quality of an image. This approach involves comparing the structural information of a possibly distorted image with that in its reference image. The extraction of the structural information that is important to our visual system is a challenging task. A sparse representation-based image quality assessment approach is proposed to address this issue. When compared with seven existing metrics, our method performs the best in three databases and ranks among the top three in the remaining three databases.

Preface

This dissertation presents the research conducted by Tanaya Guha, in collaboration with mainly Prof. Rabab Ward. Below is a list of the scientific articles written by Guha during the course of her doctoral studies at the University of British Columbia, Vancouver, Canada.

J1 T. Guha and R. K. Ward, "Image similarity using sparse representation and compression distance", submitted 2013.

J2 T. Guha, E. Nezhadarya and R. K. Ward, "Sparse representation-based image quality assessment", submitted 2013.

J3 T. Guha and R. K. Ward, "Learning sparse representations for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, issue 8, pp. 1576 - 1588, Aug 2012.

C1 T. Guha, R. K. Ward and T. Aboulnasr, "Image similarity measurement from sparse reconstruction errors," in proceedings of *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Vancouver, BC, May 2013.

C2 T. Guha and R. K. Ward, "Sparse representation-based image similarity measurement," in proceedings of *Grace Hopper Conference*, Baltimore, MD, Oct 2012.

C3 T. Guha and R. K. Ward, "A sparse reconstruction-based algorithm for image and video classification," in proceedings of *IEEE International Conference*

on Acoustics Speech and Signal Processing (ICASSP), pp. 3601-3604, Kyoto, Japan, Mar 2012.

C4 T. Guha and R. K. Ward, "Action recognition by learning class-specific overcomplete dictionaries," in proceedings of *IEEE International Conference on Face and Gesture Recognition (FG)*, pp. 143 - 148, Santa Barbara, CA, Mar 2011.

J1, J3, C2, C3, C4: Guha is the primary author and the main contributor of these articles. She developed the algorithms, implemented them, and performed the experiments and analysis. Prof. Ward suggested research ideas and directions, provided technical feedback, and contributed to the writing and presentation of the work.

J2: Guha is the primary author and the main contributor of this article. She developed the main idea which was then improved by suggestions and ideas from Dr. Nezhadarya and Prof. Ward. Guha implemented the algorithm, and performed the evaluations. Dr. Nezhadarya provided technical feedback and suggestions. Prof. Ward provided technical feedback, and suggestions on the writing and presentation of the work.

C1: Guha is the primary author and the main contributor of this article. She proposed the idea, developed and implemented the algorithm. Prof. Ward provided technical feedback, and suggestions on the writing and presentation of the work. Prof. Aboulnasr provided editorial comments.

Table of Contents

List of Tables

List of Figures

Glossary

BP	Basis Pursuit
BPDN	Basis Pursuit Denoising
CC	Pearson Linear Correlation Coefficient
DCT	Discrete Cosine Transform
EFVC	Error Feature Vector-based Clasification
HOG	Histogram of Gradients
HVS	Human Visual System
ICA	Independent Component Analysis
IFC	Information Fidelity Criterion
IWSSIM	Information Weighted Structural Similarity
KROCC	Kendall's Rank Order Correlation Coefficient
K-SVD	K-Singular Value Decomposition
LDA	Linear Discriminant Analysis
LMP	Local Motion Pattern
LOG	Laplacian of Gaussian
MP	Matching Pursuit

MAD Most Apparent Distortion

MAE Mean Absolute Error

MSE Mean Squared Error

NN Nearest Neighbor

OMP Orthogonal Matching Pursuit

PSNR Peak Signal to Noise Ratio

RANSAC Random Sample Consensus

RMS Root Mean Squared Error

RP Random Projection

RSR Random Sample Reconstruction

SSIM Structural Similarity Index

V1 Primary Visual Cortex

PCA Principal Component Analysis

PHVS-M PSNR with HVS properties

SIFT Scale Invariant Feature Transform

SPARQ Sparse Representation-based Quality

SROCC Spearmans Rank Order Correlation Coefficient

SVD Singular Value Decomposition

SVM Support Vector Machine

VIF Visual Information Fidelity

VQEG Video Quality Expert Group

VSNR Visual Signal to Noise Ratio

Acknowledgments

My journey as a PhD student at UBC would not have been possible without the help and kind support of a number of people. I take this opportunity to express my gratitude and thankfulness to them.

I am indebted to my PhD thesis advisor, Prof. Rabab Ward, for things too many to list. She has been extremely kind to me and supportive of my research. Her help, advice and guidance have been and will always be immensely valuable to me.

Sincere thanks to my thesis committee members, Prof. Jane Wang, Prof. Purang Abolmaesumi and Prof. Karthik Pattabiraman for their insightful comments and suggestions. I also extend my thanks to all those anonymous reviewers who read my journal manuscripts. Their criticism and advice have helped me improve the quality of my work.

Just when I started thinking that I am too old to make great friends, I met Mani Malek Esmaeili and Simon Fauvel. I thank them from the bottom of my heart for being such great friends over the years.

I am thankful to Ehsan Nezhadarya for being a friend and a collaborator. Thanks to Joyce Chiang for her help with my oral presentation. Ali Amiri also deserves special thanks for directing me to interesting papers and softwares from time to time. Thanks to Mona Rahmani, Zahra Ahmadian, Di Xu and all the former and

present members of ICICS X310 I have known.

I feel lucky to have made some amazing friends even outside UBC who made Vancouver feel like home. So thank you Anuradha Mitra and Santanu Mitra for your love, encouragement and help. Many thanks to Ananya Sanyal, Paulami Das and Samita Chakraborty.

Back in India, my childhood friend Payel Sengupta has been a source of continuous support and love, I can never thank her enough.

My family in India has been the real powerhouse of love and encouragement. My parents and my brother have always stood by me through the thick and thin, and my mother-in-law has been my biggest supporter. I feel blessed.

My husband, Anirban Guha, has been instrumental to this journey. I could never reach this milestone without his endless love, support and encouragement.

Chapter 1

Introduction

"Begin at the beginning," the King said, very gravely,"and go on till you come to the end; then stop."

— Lewis Carroll, *Alice in Wonderland*

1.1 Motivation

Sparse representation of signals has emerged as a major area of research in the signal and image processing community. This is because sparse analysis allows us to capture the essential information in a signal into a small number of components. This is an elegant and efficient way to deal with high dimensional signals and is useful in their processing, transmission and storage.

It is well known that many natural signals (e.g. image, video, music) can be represented sparsely when decomposed using a set of *properly chosen* basis functions. This set of bases is commonly known as a *dictionary*. A dictionary can be *complete* or *overcomplete* depending on whether the number of bases it contains

is the same or greater than the dimension of the given signal it represents. This thesis concentrates on the overcomplete case. An overcomplete dictionary offers greater flexibility in representing the essential structures in a signal which results in higher sparsity in the transform domain. The sparse representations obtained using an overcomplete dictionary are also shown to be robust to additive noise and occlusion [1].

Traditionally, researchers have used predefined basis functions (e.g. sinusoids, wavelets, curvelets) to create a dictionary. However, the success of a predefined dictionary is limited by how suitable its bases functions are to the structures in the given signal. A more generalized approach is to *learn* the dictionary from the signal itself. This data-dependent approach alleviates the difficulty of selecting the proper predefined dictionary which often requires multiple trials, experience and even mathematical analysis.

Learnt dictionaries have been shown to outperform predefined dictionaries in several signal processing tasks such as denoising, compression and reconstruction [2, 3]. It is also known that overcomplete dictionaries when learnt with a sparsity prior generate basis functions qualitatively similar to the receptive field of the simple cells in the Primary Visual Cortex (V1) [4, 5]. Motivated by the success of the learnt dictionaries and their connection to the Human Visual System (HVS), this thesis explores how useful such learnt dictionaries and their corresponding sparse representations would be for applications that require compatibility with human perception of vision. These applications are (i) classification of image and video signals, (ii) measurement of similarity between two images, and (ii) assessment of perceptual quality of an image.

1.2 The theory of sparse representation

Sparse representation is a rapidly growing field of research lying at the intersection of signal processing, applied mathematics and statistics. In this section, we briefly discuss the major concepts of sparse representation related to our interest. For more information on this topic, we request the reader to refer to the review paper by Bruckstein et al. [3] and the references therein.

We begin with the basic assumption that a signal b can be represented in terms of a linear superposition of m basis functions, $\phi_1, \phi_2, ..., \phi_m$ mixed together with coefficients $x_1, x_2, ..., x_m$.

$$b = \sum_{i=1}^{m} x_i \phi_i \tag{1.1}$$

For convenience, we adopt a matrix-vector notation, such that we have the following linear system.

$$\mathbf{b} = \Phi \mathbf{x} \tag{1.2}$$

where the signal $\mathbf{b} \in \mathbb{R}^n$, and the matrix $\Phi \in \mathbb{R}^{n \times m}$ has m basis vectors $\phi_1, \phi_2, ..., \phi_m$ as its columns, and $\mathbf{x} \in \mathbb{R}^m$ is the coefficient vector. When the majority of coefficients in $\mathbf{x}$ are zero, it is said that $\mathbf{b}$ has a *sparse representation* with respect to the *dictionary* Φ. In practice, however, it is unlikely to have the majority of coefficients exactly equal to zero. It is more common to have a representation where a few coefficients are significantly large while the rest are very small. Signals yielding such representations are thus called *compressible*, as they are not truly sparse. Nevertheless, the smaller coefficients can be safely ignored and set to zero in order to secure a *sparse* representation in many practical situations.

It is well-known that many natural signals such as audio, still images and videos

can be sparsely represented when decomposed using a properly chosen Φ. For example, music signals are sparse in the Fourier (sinusoid) bases and many natural images have sparse representation in the wavelets bases. Sparsity plays such an important role in signal representation, storage, transmission and reconstruction that the primary objective of many signal processing problems is to obtain the sparse representation of a given signal. This often comes down to solving Equation 1.2 such that $\mathbf{x}$ has as fewer non-zero components as possible.

1.2.1 Overcompleteness and the sparsest solution

Solving Equation 1.2 is easy when Φ is a known square, orthonormal matrix (i.e. when $n = m$ and $\langle \phi_i, \phi_j \rangle = 0$, $i \neq j$ and $\|\phi_i\| = 1$), such as the case in Fourier matrix. Although solving such a system is easy, the resulting $\mathbf{x}$ may or may not be sparse. It is well-known that smooth signals are sparsely representable in the Fourier domain but signals with sharp edges require a large number of Fourier terms in order to be approximated reliably (Gibbs phenomenon).

In practice, signals are often found to contain mixed structures that can not be efficiently captured by using only sinusoids or only wavelets. This leads to the idea of combining multiple bases to create an *overcomplete* dictionary - where the number of basis vectors (not necessarily orthogonal) is greater than the dimensionality of the input signal. Note that, in the overcomplete case Φ is a rectangular matrix, having more columns than rows ($n < m$) and a full-rank, overcomplete Φ makes Equation 1.2 an *underdetermined* system having infinite number of solutions. To narrow down the choice to one well-defined solution, additional constraints are required. A familiar way to do this is to introduce an objective function $J(\mathbf{x})$ and

define a general optimization problem, P_J, as follows.

$$(P_J): \quad \min_{\mathbf{x}} \quad J(\mathbf{x}) \quad \text{subject to} \quad \mathbf{b} = \Phi\mathbf{x} \tag{1.3}$$

$J(\cdot)$ is expected to be a well-behaved convex function so as to guarantee a unique solution. The most common case is where $J(\mathbf{x})$ is a measure of the energy of $\mathbf{x}$ i.e. $J_2(\mathbf{x}) = \|\mathbf{x}\|_2^2$. This generates a so-called *minimum norm solution* of the system, which has the following closed form solution.

$$\mathbf{x} = \Phi^+\mathbf{b} = \Phi^T(\Phi^T\Phi)^{-1}\mathbf{b} \tag{1.4}$$

However, we are interested in a different objective function, $J_0(\mathbf{x})$, that uses *sparsity* instead of energy. A straightforward measure of sparsity is provided by the ℓ_0 semi-norm, which counts the number of non-zero components in a vector. Let $\#i$ be the number of non-zero components in $\mathbf{x}$, then the ℓ_0 norm of $\mathbf{x}$ can be written as

$$\|\mathbf{x}\|_0 = \#i\{i : x_i \neq 0\} \tag{1.5}$$

We define $J_0(\mathbf{x}) = \|\mathbf{x}\|_0$ to form the following optimization problem (P_0).

$$(P_0): \quad \min_{\mathbf{x}} \quad \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{b} = \Phi\mathbf{x} \tag{1.6}$$

Although Equation 1.6 does not look very complicated, the difficulty of solving this equation is enormous. This is a classic case of combinatorial search: one needs to form all possible combinations of columns chosen from the m columns of Φ, generate corresponding subsystems of linear equations and verify if the subsystem

is solvable, in each case. The complexity of combinatorial search is exponential and it has been proved that solving (P_0) (Equation 1.6), in general, is NP-hard [6].

Instead of working with the exact case $\mathbf{b} = \Phi\mathbf{x}$, the constraint is often relaxed using a quadratic penalty function $\|\mathbf{b} - \Phi\mathbf{x}\|_2^2$ and the following error-tolerant versions of (P_0) is solved.

$$(P_0^\delta): \quad \min_{\mathbf{x}} \quad \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{b} - \Phi\mathbf{x}\|_2^2 \leq \delta \tag{1.7}$$

The most natural and intuitive interpretation of Equation 1.7 is to account for the noise present in the real data. Theoretical studies have defined the conditions at which the error-tolerant versions yield stable and fairly accurate solutions [7, 8].

1.2.2 Greedy algorithms

Although a straightforward approach to solving (P_0) Equation 1.6 or (P_0^δ) Equation 1.7 seems futile, there exist greedy strategies that can work under certain conditions. Instead of an exhaustive search, a *greedy strategy*, looks for a series of locally optimal single-term updates. At each iteration, a greedy algorithm selects only one column - the column of Φ that minimizes the residual error $||\mathbf{b} - \Phi\mathbf{x}||_2$. A new column is added at each iteration so as to reduce the residual error further. A pseudocode for the greedy strategy is presented in Algorithm 1.1.

A number of variants of this strategy exist in the literature [9–13]. Such algorithms are more computationally efficient than the exhaustive search, but can also fail badly in certain cases [14]. Nevertheless, greedy algorithms are extensively used in many fields under different names; the popular ones are known as the Matching Pursuit (MP) [9] and the Orthogonal Matching Pursuit (OMP) [12] in

Algorithm 1.1 Greedy algorithm

Input: $\mathbf{b}$, Φ Output: $\mathbf{x}$

- Initialize:
  - $\mathbf{x}$ = zero vector
  - *residual* $= \mathbf{b}$
- Loop until *residual* $\leq \delta$ (or 0 for the exact case)
  - Find $\phi_i \in \Phi$ with maximum inner product $|\langle \mathbf{b}, \phi_i \rangle|$
  - $x_i = \langle \mathbf{b}, \phi_i \rangle$
  - *residual* $\leftarrow ||\mathbf{b} - \Phi\mathbf{x}||_2$

the context of signal processing. For cases when a sufficiently sparse solution is known to exist, the greedy algorithms can solve (P_0) exactly [15].

1.2.3 Convex relaxation

Apart from the greedy strategy, another way of solving (P_0) is to replace the highly discontinuous ℓ_0 norm with its closest convex approximation i.e. the ℓ_1 norm.

$$(P_1): \quad \min_{\mathbf{x}} \quad \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{b} = \Phi\mathbf{x} \tag{1.8}$$

Since the problem (P_1) above is convex, it can be solved by standard optimization tools like linear programming. This approach is known as the Basis Pursuit (BP) [16]. BP and its variants are more sophisticated compared to the greedy algorithms, since they find the global solution of a well-posed optimization problem. The error-tolerant version of (P_1) in Equation 1.8, known as the Basis Pursuit

Denoising (BPDN) [17], is defined as

$$(P_1^{\delta}): \quad \min_{\mathbf{x}} \quad \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{b}-\Phi\mathbf{x}\|_2^2 \leq \delta \tag{1.9}$$

There also exist other convex relaxation techniques such as FOCUSS [18] and iterative shrinkage [3].

1.3 The idea of dictionary learning

It is now well-established that many natural signals, such as images, videos and music signals, can be represented sparsely if decomposed using a properly chosen dictionary. However, selecting the proper dictionary is not an easy task. It often requires many trials, domain knowledge, previous experience and even mathematical analysis. A more generalized idea is to *learn* the basis elements in a dictionary from the given data itself. This process of learning a dictionary also an interesting connection to the HVS that indicates sparse representation to be a probable strategy employed by the HVS (HVS is discussed in Appendix A).

1.3.1 Dictionary learning and the HVS

The visual cortex in the human brain has evolved over millions of years analyzing visual information from natural scenes and environment. Given the limited physical resources of a human brain, it is only reasonable to believe that in the struggle for existence, the cortex has discovered efficient coding techniques for representing natural images and scenes. In other words, the cortex employs certain strategies that reduce the redundancy of an image such that relatively fewer neurons are active at a particular time. This is indeed a *sparse* coding strategy.

In the field of signal and image processing, *sparsity* in signal representation has long been an important goal for many practical problems. For data storage, transmission or reconstruction, having to deal with fewer coefficients is computationally more efficient and of course, more convenient. In the pursuit of efficient (sparse) signal representation, many mathematical transforms and corresponding basis functions have been designed.

In 1996, Olshausen and Field proposed a rather interesting viewpoint [4, 5]. Instead of using the existing mathematical transforms that could sparsely represent the input signal, they proposed to design a set of basis elements from the input itself i.e. to learn them from the input signal. They enforced (i) a *sparsity prior* - an assumption that it is possible to describe the input using a small number of basis elements and (ii) *overcompleteness* - the number of basis elements is greater than the vector space spanned by the input. They showed that this strategy results in a set of basis elements that are localized, oriented, and of bandpass nature, and therefore resemble the properties of simple cells in V1 (see Section A.2.1).

Due to overcompleteness, the basis functions are non-orthogonal, and the input-output relation deviates from being purely linear. The justification of deviating from a strictly linear approach is to account for a weak form of nonlinearity exhibited by the simple cells themselves [4].

1.3.2 Dictionary learning

The discovery by Olshausen and Field, promotes the idea of *learning an overcomplete dictionary* i.e. learning a set of overcomplete basis functions from the given data [4, 5]. This idea apparently mitigates the difficulty of selecting the right basis function that would lead to a sparse representation of a given signal. An overcom-

plete dictionary can be formed i) by combining multiple orthogonal bases (such as the Identity and Fourier matrices) or ii) by selecting one of the predefined overcomplete bases, such as curvelets or bandlets [19]. However, the success of an overcomplete dictionary with predefined bases is often limited by how suitable its basis functions are in representing the structure in the signal under consideration. The dictionary learning approach, on the other hand, is more generalized as its basis vectors could be adapted to fit the structures in the given data.

This promising idea however was not exploited to its full strength until recently, primarily due to the computational difficulty in obtaining a sparse solution in the overcomplete case. Earlier approaches to learning overcomplete dictionaries [1, 20] consider the dictionary as a probabilistic model of the observed data. These methods have successfully shown that an overcomplete set of bases yield a better approximation of the underlying statistical distribution of the data and can lead to a more compact representation.

Thanks to the recent progress and the growing interest in the areas of sparse optimization, dictionary learning has become an important topic of research in the last few years. Several practical dictionary learning algorithms have now been developed [21–25]. These methods have been shown to outperform prespecified dictionaries like wavelets and produce state-of-the-art results in several real world applications such as in image and video denoising [2] and color image restoration [26].

The idea of fitting bases to a particular data distribution however is not new to the signal processing community. The well-known Principal Component Analysis (PCA) learns orthogonal bases from a particular data by finding the directions in the data with the largest variance - the principal components. An extension of

PCA, called the Independent Component Analysis (ICA), allows the learning of nonorthogonal bases from the data. In both cases, however, the bases are *complete* - the number of basis functions is equal to the dimension of the input. PCA assumes a Gaussian distribution of the data and thus can fail badly when the real distribution is non-Gaussian in nature. With ICA, the non-Gaussian distribution can be handled better, but there are distributions which can not be modeled efficiently by PCA nor ICA.

Let $\mathbf{B}$ be a matrix containing s number of training samples as its columns: $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^{s}$, $\mathbf{b}_i \in \mathbb{R}^n$. The problem is to learn a dictionary $\Phi \in \mathbb{R}^{n \times m}$ ($n < m$) such that each $\mathbf{b}_i$ has a sparse representation $\mathbf{x}_i$. This can be formally expressed as the following optimization problem:

$$\min_{\Phi,\mathbf{X}} \quad \sum_{i}^{s} \|\mathbf{x}_i\|_0 \quad \text{subject to} \quad \|\mathbf{B} - \Phi\mathbf{X}\|_F^2 \leq \delta \tag{1.10}$$

where $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{s}$, $\mathbf{x}_i \in \mathbb{R}^m$ and $\|\cdot\|_F$ denotes the *Frobenius* matrix norm (the ℓ_2 norm of the vector obtained by concatenating the columns of the matrix into a single vector).

Notice that, the problem is more ill-posed compared to Equation 1.7 or Equation 1.9 because of the two unknowns: Φ and $\mathbf{X}$. Usually this is solved iteratively by performing two steps at each iteration - keep Φ fixed and solve for $\mathbf{X}$; next, update Φ according to the new $\mathbf{X}$. Various methods [21, 23, 2, 24, 27] have proposed different ways of performing the sparse optimization and the corresponding updates, each having its own merits and demerits.

1.4 Applications of sparse representation

The earlier research effort in sparse representation is mainly concentrated on developing the theory of sparse representation as a new paradigm in signal processing. Later sparse signal analysis found applications in those problems that involve signal recovery, such as in denoising, compression, signal restoration and reconstruction [3].

One of the more successful real-world applications is compressed sensing-based dynamic magnetic resonance image reconstruction [28]. Applications of sparse analysis have also been extended to remote sensing, audio signal processing, geophysical data analysis, computational biology and other areas. An extensive list of applications (and theoretical developments) involving sparse analysis is maintained by Rice University, Texas [29].

Since the last few years the area of sparse representation has become one of the most active areas in signal and image processing. Other than denoising, restoration and inpainting, there are a number of tasks, such as encryption, watermarking, scrambling and target detection, that can benefit from sparse analysis. Recently, sparse representation has been used to address classification problems such as face recognition [27], object recognition [30], texture classification [31], etc.

1.5 Objective

This thesis explores the application of sparse representations obtained by learning overcomplete dictionaries for three types of application areas that require compatibility with human visual perception. These applications involve

- *Classification of image and video signals* information relevant to human per-

ception, such as the type of object, scene, activity or identity of a person, needs to be extracted from given data.

- *Measurement of similarity between two images* is a fundamental issue and of critical importance in many applications. This problem often demands compatibility with human intuition.

- *Assessment of perceptual quality of an image* can be seen as a special case of image similarity measurement where the goal is to quantify image degradations as perceived by humans. While the general purpose image similarity measures concentrate on achieving robustness against translation, rotation, noise and other distortions, quality measures attempt to quantify those distortions.

1.6 Contributions

This main contributions of this thesis lie in extending and enriching the application of sparse representations obtained by learning overcomplete dictionaries. Below, we list the broad contributions of this thesis (the contributions are discussed in Chapter 5 in detail).

- This thesis is one of the pioneering works that explore the application of learning sparse representations for classification. We have proposed four classification algorithms. These algorithms are applied to a variety of classification problems: face recognition, human action recognition, expression recognition and biological species classification. The proposed algorithms consistently perform better or at par with the state-of-the-art.

- The application of learnt sparse representations have been extended to two new areas: image similarity measurement and perceptual quality assessment of an image.

- A sparse representation-based image similarity measure has been proposed. This measure extends the current state-of-the-art in generic similarity measure based on the idea of data complexity. The proposed measure produces successful results for image clustering, classification and retrieval.

- A new image quality assessment metric has been proposed. This metric outperforms a number of well established quality metrics and performs better or at par with the current state-of-the art.

1.7 Organization

The rest of this thesis is organized as follows:

Chapter 2 presents three classification algorithms developed under different dictionary learning frameworks following the supervised classification paradigm. This chapter concentrates on the problems of recognizing faces and human actions in videos. The proposed algorithms are evaluates on benchmark datasets.

Chapter 3 proposes a generic similarity measure that is applicable to a variety of problems and data. The generality and effectiveness of our measure is demonstrated in the context of clustering, retrieval and classification. This chapter also connects sparse representation with the ideas of compression-based distance and Kolmogorov complexity.

Chapter 4 develops an image quality assessment method. The proposed method is based on the idea of learning sparse representation of images. Our method is

evaluated on six publicly available image quality evaluation datasets and is shown to perform better or at par with the state-of-the-art.

Finally, Chapter 5 concludes the thesis, listing the contributions and future work for each part separately.

Chapter 2

Sparse Representation-based Classification

In this chapter, we explore the effectiveness of sparse representation obtained by learning overcomplete dictionaries in the context of classifying images and videos. We investigate three dictionary learning frameworks. For each framework, we develop one or more classification algorithms. These frameworks and algorithms are fairly general and are applicable to a variety of classification problems. In this chapter, we present results for two challenging problems: image-based face recognition and video-based human action recognition. Additional results are provided in the Appendix.

2.1 Background and motivation

The theory of sparse representation is developed primarily to address the problems like signal denoising, reconstruction and compression. Recently, it has been shown that sparse representation can also be useful in addressing classification problems.

This is because sparse representation is *naturally discriminative* - it selects from many basis vectors, only those that most compactly represent a signal [27].

The success of the sparse reconstruction-based classification algorithms largely depends on the choice of the dictionary. As mentioned in Chapter 1, predefined dictionaries such as curvelets, bandlets and variants of wavelets can be used. But the success of these dictionaries depends on their suitability in capturing the structures in the signal under consideration.

Another approach to building a dictionary is by concatenating the vectorized training samples of all classes together [27]. This approach is successfully used in face recognition. However, constructing such a dictionary requires a good number of training samples to be available for each class. This may not always be the case in practice.

A more generalized approach to designing a dictionary is to learn the dictionary from a set of training data. Such a dictionary learning approach has been employed in texture classification and segmentation [31]. A recent work proposes learning a dictionary by jointly optimizing an energy formula containing both sparse reconstruction and class discrimination components [24]. This work reported preliminary results on image segmentation. However, the joint optimization approach proposed in [24] introduces further difficulty to the already complicated optimization task.

A work on object recognition [30] moved from pixel domain to feature domain and obtained sparse decomposition of the *Scale Invariant Feature Transform (*SIFT*)* features [32]. This method creates a dictionary of SIFT features using sparse coding, but sticks to the traditional *Support Vector Machine (*SVM*)* for classification.

The area of sparse representation-based classification, though rapidly growing,

is still at an early stage. Prior work on classification using sparse representation has mainly dealt with images. Videos, being functions of space and time, pose a bigger challenge. There also exist the need for developing more efficient and discriminative classification frameworks. The work that we present in this chapter, explores the usefulness of sparse representation obtained using learnt dictionaries in the context of image-based face recognition and video-based human action recognition.

2.2 Proposed approach

Our approach follows the supervised classification paradigm where the availability of a labeled training dataset is assumed. Let us consider a labeled dataset of images or videos having K different classes of data. Let the available training samples per class be l. The training samples are represented as $\mathbf{V}_{ij}$, $i = 1,2,...,K$ and $j = 1,2,...,l$.

In our approach, the first step is to extract suitable features from the available training data. The dimensionality of those features is reduced if necessary. Overcomplete dictionaries are then learnt from the lower-dimensional features. When a new query data is available, similar features are extracted from the query. The classification algorithm uses these features to assign the query to the proper class.

2.2.1 Feature extraction

The first step is to extract proper features from the training data. For images, this step can be as simple as *randomly* extracting raw patches from the images. Popular feature extraction methods such as SIFT [32] can also be employed to extract a set of meaningful features. In this work, we use randomly extracted image patches as

features for the image-based classification task.

A common approach to obtaining a rich representation of a video sequence is to extract a set of local, spatio-temporal features. We choose to extract the (a) *Cuboid* [33] features and (b) the newly designed Local Motion Pattern (LMP) features [34] from the training video sequences. The Cuboids method is chosen because of its wide popularity in the field of action recognition. The LMP offers a fast technique to extract spatio-temporal features from videos. LMP has been developed by the author of this thesis. Both of the feature extraction methods can generate a good number of features from a video, which is an important requirement for learning the dictionary of features. A brief description of the Cuboid features is given below. A description of the LMP features can be found in Appendix C.

In order to extract the Cuboid features, the key points in a video sequence need to be detected. These key points are detected by applying separable linear filters to the video sequence. A response function is computed by convolving the video sequence with a 2D Gaussian filter (applied only in the spatial domain) and a quadrature pair of 1D Gabor filters (applied in the temporal direction). The Gaussian and the Gabor filter contain parameters to control the spatial and temporal scales. The local maxima points of the response function are detected as the key points. A small video patch is extracted around each of the key points and is converted to a 1D feature vector. There are a number of ways to compute such a feature vector from the video patch [33]. Among those, gradient-based descriptors like Histogram of Gradients (HOG) and concatenated gradient vectors are the most reliable ones [33]. For more details about the Cuboid features refer to the original work [33].

2.2.2 Dimensionality reduction

Assume that the number of feature vectors extracted from a training sample (image or video) is s. Let each vector be d-dimensional i.e. $\mathbf{d}_i \in \mathbb{R}^d$. Then the set of features can be denoted as $\mathbf{D} = \{\mathbf{d}_i\}_{i=1}^{s}$, $\mathbf{d}_i \in \mathbb{R}^d$.

As d can be very large, these features are typically high-dimensional. This high dimensionality seriously limits the speed and practical applicability of these features. A natural solution is to reduce the dimensionality. The application of standard methods like PCA and Linear Discriminant Analysis (LDA), to obtain lower dimensional representation, is well-known. Recently, Random Projection (RP) has emerged as a powerful tool in dimensionality reduction [35]. Theoretical results show that the projections on a random lower-dimensional subspace can preserve the distances between vectors quite reliably. The advantages of RP are that it is data-independent, simple and fast.

The original d-dimensional descriptors are projected onto an n dimensional subspace ($n << d$) by premultiplying the descriptor matrix $\mathbf{D} \in \mathbb{R}^{d \times s}$ by a random matrix $\mathbf{R} \in \mathbb{R}^{n \times d}$. In practice, any normally distributed $\mathbf{R}$ with zero mean and unit variance serves the purpose (choices of non-Gaussian random matrices are also available). The dimensionality reduction step is hence a simple matrix multiplication, given by

$$\mathbf{B} = \mathbf{RD} \tag{2.1}$$

where the reduced data matrix $\mathbf{B} \in \mathbb{R}^{n \times s}$ contains projections (not true projections, because the vectors are not orthogonal) of $\mathbf{D}$ on some random n dimensional subspace.

2.2.3 Dictionary learning

The next step is to learn the overcomplete dictionary (or dictionaries) of the features and their corresponding sparse representation. We start with briefly describing the dictionary learning algorithm employed in this work.

Consider a set of lower dimensional features $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^{s}$, $\mathbf{b}_i \in \mathbb{R}^n$. We wish to learn a dictionary $\Phi \in \mathbb{R}^{n \times m}$ ($m > n$) such that each vector $\mathbf{b}_i \in \mathbf{B}$ has a sparse representation $\mathbf{x}_i$ w.r.t. Φ. Each $\mathbf{x}_i \in \mathbb{R}^m$ is a sparse vector i.e. $\mathbf{x}_i$ contains k ($k << n$) or fewer non-zero elements. This can be formally expressed as the Equation 1.10 or the following equivalent optimization problem.

$$\min_{\Phi, \mathbf{X}} \quad \|\mathbf{B} - \Phi\mathbf{X}\|_F^2 \quad \text{subject to } \forall i \quad \|\mathbf{x}_i\|_0 \leq k \tag{2.2}$$

where $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{s}$, $\mathbf{x}_i \in \mathbb{R}^m$.

To solve Equation 2.2, a recently developed dictionary learning algorithm, known as the K-Singular Value Decomposition (K-SVD) [23] is used. K-SVD iteratively solves Equation 2.2 by performing two steps at every iteration: (i) sparse coding and (ii) dictionary update. In the sparse coding step, Φ is kept fixed and $\mathbf{X}$ is computed.

$$\min_{\Phi} \quad \|\mathbf{B} - \Phi\mathbf{X}\|_F^2 \quad \text{subject to } \forall i \quad \|\mathbf{x}_i\|_0 \leq k \tag{2.3}$$

K-SVD uses the greedy algorithm OMP to solve the Equation 2.3 approximately.

In the dictionary update step, the atoms of the dictionary Φ are updated sequentially, allowing the relevant coefficients in $\mathbf{X}$ to change as well. Updating an atom $\phi_i \in \Phi$ involves computing a rank-one approximation of a residual matrix as

follows.

$$\mathbf{E}_i = \mathbf{B} - \widetilde{\Phi_i}\widetilde{\mathbf{X}_i} \tag{2.4}$$

where $\widetilde{\Phi_i}$ and $\widetilde{\mathbf{X}_i}$ are formed by removing the i-th column from Φ and the i-th row from $\mathbf{X}$. This rank-one approximation is computed by subjecting $\mathbf{E}_i$ to a *Singular Value Decomposition (*SVD*)*. For detailed description of K-SVD algorithm refer to the original work [23, 2].

Recall that our dataset contains training samples from K different classes. For each class, a set of feature vectors is extracted from each of the training sequences. In order to construct a dictionary from these features, we consider the three dictionary learning options listed below:

- *shared* dictionary - learning a single dictionary for all classes.
- *class-specific* dictionaries - learning K dictionaries, one for each class.
- *concatenated* dictionaries - a single dictionary formed by concatenating the K class-specific dictionaries

Shared dictionary-based classification

In this framework, a single shared dictionary Φ_S is learnt for all K classes, so that multiple classes can share some common dictionary elements. Since the dictionary is learnt only once, it saves on the computations. But in this case, a bigger dictionary might be needed to accommodate the variations of all classes. The learning process also has to be repeated whenever a new class is added to the system.

Let the matrix $\mathbf{B}$ contain all the features extracted from the training samples of *all classes*, and let $\mathbf{X}$ contain their corresponding sparse representations w.r.t. the

Algorithm 2.1 Classification using shared dictionary

- Learn a single shared dictionary Φ_S as in (4.1).
- Compute the coefficient histograms $\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_K$, one for each class using (2.5).
- Find the sparse representation of the query features $\mathbf{Q}$:

$$\min_{\mathbf{X}_Q} \quad \|\mathbf{Q} - \Phi_S \mathbf{X}_Q\|_F^2 \quad \text{subject to} \quad \|\mathbf{x}_{Qi}\|_0 \leq k$$

- Compute the histogram $\mathbf{h}_Q$ pertaining to $\mathbf{Q}$.
- Estimate query class:

$$\hat{i}_Q = \underset{i \in 1,2,...K}{\operatorname{argmax}} \ \mathbf{h}_Q^T \mathbf{h}_i$$

shared dictionary Φ_S. The sparse coefficients in a column vector $\mathbf{x}_i \in \mathbf{X}$ indicate the contribution of each of the dictionary atoms in approximating the feature $\mathbf{b}_i \in \mathbf{B}$. Thus the sparse coefficients corresponding to all the descriptors of a particular class collectively demonstrate the contribution of the dictionary atoms to the representation of that class. Hence some statistics of these sparse coefficients (sometimes called descriptors codes) will be able to characterize that class. A popular statistical representation is the histogram of coefficients. Let the J-th class have a sparse decomposition $\mathbf{X}_J = \{\mathbf{x}_{Ji}\}_{i=1}^s$ over Φ_S. Then its histogram of coefficients $\mathbf{h}_J$ is computed as follows.

$$\mathbf{h}_J = \frac{1}{s} \sum_{i=1}^{s} \mathbf{x}_{Ji} \tag{2.5}$$

Given a query video sequence $\mathbf{V}_Q$, it is represented by a set of features $\mathbf{Q} = \{\mathbf{q}_i\}_{i=1}^r$, $\mathbf{r}_i \in \mathbb{R}^n$. The shared dictionary Φ_S is used to determine the class of $\mathbf{V}_Q$. The pseudocode for this shared dictionary-based classification algorithm that uses

the class histograms is presented in Algorithm 2.1.

Class-specific dictionary-based classification

This framework learns K dictionaries, $\Phi_1, \Phi_2, ..., \Phi_K$, one for each class. One advantage of having class-specific dictionaries is that each class is modeled *independently* of the others and hence painful repetition of the training process when a new class of data is added to the system is no longer necessary. This also indicates the possibility of parallel implementation.

The basic idea is that a dictionary tailored to represent one particular class of action will have an *efficient* representation of this class and at the same time will be *less efficient* in representing the actions belonging to a different class. The *efficiency* here refers to the lower reconstruction error while sparsity is constant. We exploit this inherent discriminative nature of the class-specific dictionaries and develop the two following classification techniques:

1. Random Sample Reconstruction (RSR)

2. Error Feature Vector-based Clasification (EFVC)

Recall that the query data $\mathbf{V}_Q$ is represented by a collection of features as $\mathbf{Q} = \{\mathbf{q}_i\}_{i=1}^{r}$, $\mathbf{r}_i \in \mathbb{R}^n$. A simple way to classify $\mathbf{V}_Q$ is to find the K approximations of $\mathbf{Q}$ given by each of the K learnt dictionaries and to compute their corresponding reconstruction errors e_i where $i = 1, 2, ...K$.

$$e_i = \left\| \mathbf{Q} - \Phi_i \hat{\mathbf{X}}_{Q_i} \right\|_2^2 \tag{2.6}$$

where

$$\hat{\mathbf{X}}_{Q_i} = \underset{\mathbf{X}_{Q_i}}{\operatorname{argmin}} \quad \|\mathbf{Q} - \Phi_i \mathbf{X}_{Q_i}\|_F^2 \quad \text{subject to} \quad \|\mathbf{x}_{Qi}\|_0 \leq k \tag{2.7}$$

Here $\mathbf{X}_{Q_i} = \{\mathbf{x}_{Qi}\}_{i=1}^{r}$, $\mathbf{x}_i \in \mathbb{R}^m$ is the sparse representation of $\mathbf{Q}$ w.r.t. the dictionary Φ_i. Then the estimated class of $\mathbf{V}_Q$ (denoted as $\hat{i}_Q$) is the class that yields the smallest e_i.

$$\hat{i}_Q = \underset{i \in [1,2,\ldots,K]}{\operatorname{argmin}} \quad e_i \tag{2.8}$$

This method discriminates on the basis of the reconstruction errors which has been proved to be quite useful in texture classification [31, 24]. We will refer to this method as the *Simple Reconstruction* method.

1. **Random sample reconstruction (RSR)**

In a complex problem like classification, a strong presence of outliers in $\mathbf{Q}$ is highly probable due to noisy data, occlusion, errors in keypoint detection, etc. In the presence of a large number of outliers, if all the features in $\mathbf{Q}$ are used for reconstruction, the resulting reconstruction error will hardly be a reliable means of classification.

In order to build a robust classifier, we propose the idea of *Random Sample Reconstruction* (RSR). This is motivated by the celebrated Random Sample Consensus (RANSAC) algorithm [36]. The RANSAC algorithm finds the part of the data that best fits a given model, whereas the proposed RSR solves an even more difficult problem - it finds both the best model (dictionary) among a number of probable ones, and the part of the data that best fits the chosen model.

The basic assumption of the proposed RSR algorithm is that the best model (dictionary)can be estimated by a small number of good data points i.e. in our

case error-free feature vectors. Let r be the total number of features extracted from the query. Let the number of error-free features i.e. good data points be g, where $g << r$.

Let the probability of selecting one good feature be ω and the probability of observing an outlier is $(1-\omega)$. If we perform λ trials and in each trial select g random features, the probability of selecting at least one error-free set of g features is $1-(1-\omega^g)^\lambda$. We want to ensure that such a set can be selected with a probability $\mathscr{P}$. Therefore we have the following relationship.

$$1-(1-\omega^g)^\lambda = \mathscr{P} \tag{2.9}$$

For given values of $\mathscr{P}$ and ω, the value of λ that ensures the success of selecting an error-free set of features is computed as

$$\lambda = \frac{\log(1-\mathscr{P})}{\log(1-\omega^g)} \tag{2.10}$$

At every trial, a random subset of g features is selected. Let this subset be denoted as $\mathbf{Q}_g$. The best model (dictionary) for $\mathbf{Q}_g$ is estimated by the simple reconstruction method described in Section 2.2.3. The features that are not in $\mathbf{Q}_g$, are then approximated by the estimated model. The features, for which the reconstruction error is below a certain threshold, are called the *inliers*. Our algorithm eventually selects the model that has the largest number of inliers. Note that, the number of good data points g is unknown. So, for our experiments g is set to 1% of the total number of available data points i.e. $g = 0.01q$. The values of ω and λ are updated at each iteration. The proposed algorithm can determine the class only with

Algorithm 2.2 Random Sample Reconstruction (RSR)

- Initialize: No. of inliers $I_0 = 0$; total no. of data points = r; no. of good data points, $g = 0.01r$; $\mathscr{P} = 0.99$.
- Compute:

$$\omega = \frac{g + I_0}{q}$$

$$\lambda = \frac{\log(1-\mathscr{P})}{\log(1-\omega^g)}$$

- Loop until $\lambda = 0$
  - Form $\mathbf{Q}_g \subset \mathbf{Q}$ by choosing g random data points
  - Estimate the class of $\mathbf{Q}_g$ using Equation 2.6 - Equation 2.8; let the estimated class of $\mathbf{Q}_g$ be ρ and the corresponding dictionary be Φ_ρ.
  - For every $\mathbf{q}_i \notin \mathbf{Q}_g$
    - Compute:

$$\varepsilon_i = \left\| \mathbf{q}_i - \Phi_\rho \hat{\mathbf{x}}_{g_i} \right\|_2$$

$$\hat{\mathbf{x}}_{g_i} = \underset{\mathbf{x}_{g_i}}{\operatorname{argmin}} \quad \|\mathbf{q}_i - \Phi_\rho \mathbf{x}_{g_i}\|_F^2 \quad \text{subject to} \quad \|\mathbf{x}_{g_i}\|_0 \leq k$$

    - Count inliers: $I \leftarrow \{i : \varepsilon_i \leq T_h\}$ where $T_h = 0.3 * \|\mathbf{q}_i\|_2$ threshold.
  - *Update*: If $|I| > I_0$
    - set $I_0 \leftarrow |I|$
    - estimate class: $\hat{i}_Q \leftarrow \rho$
    - update ω and λ

a certain probability $\mathscr{P}$. A less conservative value of $\mathscr{P}$ can be used to achieve faster convergence. Algorithm 2.2 presents the pseudocode for the proposed RSR algorithm.

2. **Error feature vector-based classification (EFVC)**

The proposed RSR method approximates the query using each of the K class-

Algorithm 2.3 Error Feature Vector-based Clasification (EFVC)

- For each training sample $\mathbf{V}_{ij}$, $i = 1,2,...,K$ and $j = 1,2,...,m$
  - Compute K reconstruction errors $\varepsilon^1, \varepsilon^2, ..., \varepsilon^K$ using Equation 2.11
  - Construct the error vector $\mathbf{E}_{ij}$ using Equation 2.12
- Given query $\mathbf{Q}$, form $\mathbf{E}_Q = \left[\varepsilon_Q^1,\ \varepsilon_Q^2,\ \ ...\ \varepsilon_Q^K\right]^T$
- Estimate class:

$$\hat{i}_Q = \underset{i\in[1,2,..,K]}{\operatorname{argmin}} \quad dist\left(\mathbf{E}_Q, \mathbf{E}_{ij}\right)$$

  where $dist\left(\mathbf{E}_Q, \mathbf{E}_{ij}\right) = \sqrt{\left(\mathbf{E}_Q - \mathbf{E}_{ij}\right)^T \mathbf{L}\left(\mathbf{E}_q - \mathbf{E}_{ij}\right)}$, and $\mathbf{L}$ is the Mahalanobis distance matrix.

specific dictionaries and selects the dictionary that produces the minimum error. In order to increase the discriminating power, we propose to use all the reconstruction errors each training sample produces w.r.t. each dictionary, and use the corresponding errors to construct an *error feature vector*. Let the matrix $\mathbf{B} \in \mathbb{R}^{n\times s}$ contain the s descriptors pertaining to a training sample $\mathbf{V}_{ij}$. Given the class-specific dictionaries, $\Phi_1, \Phi_2, ..., \Phi_K$, $\mathbf{B}$ is approximated by each of the dictionaries to generate K corresponding reconstruction errors $\varepsilon^1, \varepsilon^2, ..., \varepsilon^K$.

$$\varepsilon^i = \sqrt{\frac{1}{s}\sum_{i=1}^{s} \left\|\mathbf{b}_i - \Phi_i \mathbf{x}_i\right\|_2^2} \tag{2.11}$$

where $i = 1,2,...,K$.

For each training sample, an error vector is constructed as follows:

$$\mathbf{E}_{ij} = \left[\varepsilon_{ij}^1,\ \varepsilon_{ij}^2,\ \ ...\ \varepsilon_{ij}^K\right]^T \tag{2.12}$$

Each training sample $\mathbf{V}_{ij}$, $i = 1,2,...,K$, $j = 1,2,...,m$ is now represented as its

corresponding error vector E_{ij} which serves as an input to a Nearest Neighbor (NN) classifier. The Mahalanobis distance metric $\mathbf{L}$ used in our algorithm is learnt using an optimization algorithm proposed in [37]. The pseudocode for the EFVC method is provided in Algorithm 2.3.

Concatenated dictionary-based classification

The third option to construct a dictionary is by concatenating the class-specific dictionaries together. A bigger dictionary Φ_C is formed by concatenating K dictionaries together.

$$\Phi_C = [\Phi_1|\Phi_2|...|\Phi_K] \tag{2.13}$$

Let us assume that originally the query data belongs to the class ρ. If $\mathbf{Q}$ is approximated by Φ_C, ideally, every $\mathbf{q} \in \mathbf{Q}$ should use the atoms of Φ_ρ only for its representation. Although this condition is difficult to achieve in practice (due to errors in $\mathbf{Q}$ and correlation among the class-specific dictionaries), we can still expect that the atoms of Φ_ρ should be used more than any other dictionary atoms. This results into a higher concentration of non-zero elements in the coefficients corresponding to Φ_ρ compared to other sub-dictionaries. The pseudocode for this classification algorithm is presented in Algorithm 2.4.

Clearly, $\mathbf{Q}$ is block sparse; this is because the non-zero coefficients in $\hat{\mathbf{X}}_Q$ occur in clusters. This encourages us to exploit block sparsity as an additional structure. But, each block in Φ_C is an overcomplete dictionary, which makes it difficult to use block sparsity promoting algorithms like block-OMP [38]. We have used block-OMP and observed that the experimental results are neither consistent nor very accurate.

Algorithm 2.4 Classification using the concatenated dictionary

- Form Φ_C as in Equation 2.13
- Find the sparse representation of the query descriptors $\mathbf{Q}$:

$$\hat{\mathbf{X}}_Q = \underset{\mathbf{X}_Q}{\text{argmin}} \quad \|\mathbf{Q} - \Phi_C \mathbf{X}_Q\|_F^2 \quad \text{subject to} \quad \|\mathbf{x}_{Qi}\|_0 \leq k$$

$$\hat{\mathbf{X}}_Q = [\mathbf{X}_{\Phi_1} | \mathbf{X}_{\Phi_2} | ... | \mathbf{X}_{\Phi_K}]$$

  where $\mathbf{X}_{\Phi_i}$ is the coefficient matrix corresponding to Φ_i.

- Estimate class:

$$\hat{i}_Q = \underset{i \in [1,2,...,K]}{\text{argmax}} \quad \|\mathbf{X}_{\Phi_i}\|_0$$

2.3 Performance evaluation

This section presents a critical evaluation of the proposed sparse representation-based classification algorithms in the context of two applications: image-based *face recognition* and *human action recognition* in videos. Experiments are performed on 1 facial image database and 2 human action databases under four settings:

- Shared dictionary with histogram correlation (*Shared-hist*)
- Class-specific dictionary using RSR (*RSR*)
- Class-specific dictionary using EFVC (*Error vector-based*)
- Concatenated dictionary (*Concat*)

2.3.1 Image-based face recognition

Face recognition experiments are performed on the *AT&T face database*. This benchmark dataset contains 400 grayscale images of 40 individuals in 10 poses.

Figure 2.1: Sample images from the AT&T face database

The images were taken at different times, with varying illumination, facial expressions and details. Each image is of dimension 92×112. Sample images from this database are presented in Figure 2.1.

For feature extraction, 1000 random patches of size 24×24 are extracted from each image. Each patch is converted to a vector of dimension 576. These high dimensional patch vectors are projected onto a random 64-dimensional subspace using RP. The shared dictionary $\Phi_S \in \mathbb{R}^{64\times256}$ is learnt using $k = 8$ and 20 KSVD iterations. Each of the class-specific dictionaries $\Phi_i \in \mathbb{R}^{64\times128}$ where $i \in [1,2,...,K]$ is learnt with $k = 8$ and 20 KSVD iterations. The concatenated dictionary is $\Phi_C \in \mathbb{R}^{64\times5120}$. The shared-hist, RSR and error vector methods use $k = 8$ in the classification stage and the concat method uses $k = 2$.

A training set is constructed by randomly selecting 7 images per class and the rest is used for testing. The results shown in Table 2.1 are the mean accuracy computed over 10 runs. At each run a new training set and a test set is constructed. The approach in [27] is a state-of-the-art sparse representation-based face recognition method. Table 2.1 shows that our proposed classification algorithms work (except shared-hist) better than the state-of-the-art. The highest recognition accuracy of 96.5% is achieved by EFVC.

Table 2.1: Comparison with state-of-the-art on the AT&T face dataset

Classification method	Recognition accuracy (%)
Eigenface	92.6
ICA	93.8
Wright et al. [27]	**94.3**
Shared-hist	91.6
RSR	94.6
EFVC	**96.5**
Concat	95.4

2.3.2 Action recognition in videos

Recognizing human actions is a key component in many applications such as human-computer interface, video surveillance, sports events, video indexing, etc. In this section, we address the problem of human action recognition in videos. Two publicly available human action databases are used in our experiments

- Weizmann action database [39]
- UCF Sports database [40]

Classification experiments are carried out separately with two different features: Cuboids and LMP. In order to extract Cuboid features from an action sequence, the Cuboid feature extraction method is applied to the video at 2 spatial and 3 temporal scales. Cuboids use the gradient based HOG descriptors in order to convert the video patches (selected around the key points) into 1D vectors. Each of the resulting Cuboid-HoG features is of dimension $[1440 \times 1]$.

To extract the LMP features, the feature extraction method are also is applied to the video at 2 spatial and 3 temporal scales. Each of the resulting LMP feature is of dimension $[1728 \times 1]$.

Figure 2.2: Sample frames from the Weizmann action dataset: bend (w1), jumping jack (w2), jump forward (w3), jump in place (w4), run (w5), gallop sideways (w6), skip (w7), walk (w8), wave one hand (w9) and wave both hands (w10).

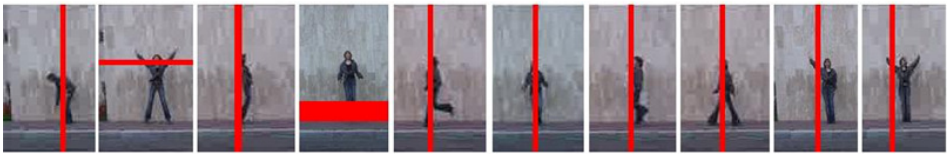


Figure 2.3: Synthetic occlusion created by the author on the same dataset.

Figure 2.4: Sample frames from the Weizmann robustness dataset showing occlusion, unusual scenarios and viewpoint variations.

The high dimensional features (Cuboids or LMP) are then projected onto a random 128-dimensional space. The shared dictionary $\Phi \in \mathbb{R}^{128\times512}$ and the class-specific dictionaries $\Phi_i \in \mathbb{R}^{128\times256}$, $i \in [1,2,...,K]$ are learnt using $k = 12$ and 20 K-SVD iterations. The shared-hist, RSR and error vector methods use $k = 12$ in the classification stage and the concat method uses $k = 2$. Note that, the theory of sparse representation and dictionary learning is in a developing stage; how to set the parameters like the optimal dictionary size and the sparsity constraint are still open issues.

	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
w1	**1**	0	0	0	0	0	0	0	0	0
w2	0	**1**	0	0	0	0	0	0	0	0
w3	0	0	**1**	0	0	0	0	0	0	0
w4	0	0	0	**1**	0	0	0	0	0	0
w5	0	0	0	0	**1**	0	0	0	0	0
w6	0	0	0	0	0	**1**	0	**0**	0	0
w7	0	0	0	0	0	0	**1**	0	0	0
w8	0	0	0	0	0	0	0	**1**	0	0
w9	0	0	0	0	0	0	0	0	**.89**	.11
w10	0	0	0	0	0	0	0	0	0	**1**

(a)

	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
w1	**1**	0	0	0	0	0	0	0	0	0
w2	0	**1**	0	0	0	0	0	0	0	0
w3	0	0	**1**	0	0	0	0	0	0	0
w4	0	0	0	**1**	0	0	0	0	0	0
w5	0	0	0	0	**1**	0	0	0	0	0
w6	0	0	0	0	0	**1**	0	**0**	0	0
w7	0	0	.11	0	0	0	**.89**	0	0	0
w8	0	0	0	0	0	0	0	**1**	0	0
w9	0	0	0	0	0	0	0	0	**.89**	.11
w10	0	0	0	0	0	0	0	0	0	**1**

(b)

Figure 2.5: (a) LMP + Concat. (mean accuracy 98.9%) and (b) LMP + RSR (mean accuracy 97.8%).

Weizmann action database

The Weizmann action database is a benchmark dataset. Since this database is frequently used by researchers, it provides a good platform for comparing the proposed approach with different action recognition approaches under similar experimental setup. The database consists of 90 low-resolution (180×144, deinterlaced 50 fps) video sequences of 9 subjects, each performing 10 natural actions: bend, jumping jack, jump forward, jump in place, run, gallop sideways, skip, walk, wave one hand and wave both hands (see Figure 2.2). The database uses a fixed camera setting and a simple background. No occlusion or viewpoint changes are present originally. Variations in spatial and temporal scale are also minimal. We have used the pre-aligned, background subtracted silhouettes provided with the original database *only* for this dataset.

The performances of the proposed algorithms in conjunction with two different features (Cuboids and LMP) are presented in Table 2.2. The lowest error rate is achieved by the concatenated dictionary when used in combination with the LMP

Table 2.2: Comparison with state-of-the-art on the Weizmann action dataset

Approach	Recognition accuracy (%)
Scovanner et al. [41]	84.2
Niebles et al. [42]	90.0
Zhang et al. [43]	92.8
Thurau & Hlavac [44]	94.4
Junejo et al. [45]	95.3
Ali & Shah [46]	95.6
Gorelick et al. [39]	**97.8**
Cuboids-based results	
Shared-hist	94.5
RSR	95.6
EFVC	91.1
Concat	95.6
LMP-based results	
Shared-hist	95.6
RSR	97.8
EFVC	95.6
Concat	**98.9**

descriptors, and the resulting recognition accuracy is 98.9% (1 misclassification out of 90). The confusion matrices corresponding to the two higher recognition results achieved in our experiments are presented in Figure 2.5.

In Table 2.2, results of the proposed algorithms are also compared with a number of existing action recognition approach. All the methods that we have compared with use the leave-one-out scheme to evaluate their respective algorithms. The proposed concat method achieves the highest recognition accuracy.

Synthetic occlusion: We also test the robustness of our classification algorithms against occlusion. Since the original dataset has no occlusion, we have selected a set of 10 action sequences from the original dataset and artificially created occlusion in all or some of the frames (see Figure 2.2, bottom row). All the four proposed

Table 2.3: Results on the Weizmann action dataset: Performance under synthetic occlusion using LMP descriptors.

Test sequence	Ground truth	Shared-hist	RSR	EFVC	Concat
occluded by a pole	bend	bend	bend	bend	bend
occluded by a bar	jack	jack	jack	jack	jack
occluded by a pole	jump	jump	jump	jump	jump
occluded feet	pjump	pjump	pjump	pjump	pjump
occluded by a pole	run	run	run	run	run
occluded by a pole	side	side	side	side	side
occluded by a pole	skip	skip	skip	skip	skip
occluded by a pole	walk	walk	walk	walk	walk
occluded by a pole	wave1	wave1	wave1	wave1	wave1
occluded by a pole	wave2	wave2	wave2	wave2	wave2

classification algorithms achieve perfect accuracy under synthetic occlusion. The results are presented in Table 2.3.

Real occlusion and viewpoint changes: There are 20 additional video sequences known as the *Weizmann Robustness dataset*, where the subjects walking in a non-uniform background create various difficult scenarios due to occlusion, clothing changes, unusual walking style and viewpoint changes. Ten of the sequences exhibit viewpoint changes and the rest contains occlusion (see Figure 2.4).

Our system is trained on the Weizmann action dataset. The video sequence from the robustness dataset are presented as the queries. Table 2.4 presents the results under occlusion and viewpoint changes. Our results are compared with the only result reported on this database i.e. with the result of Gorelick et al. [39]. RSR and Concat demonstrate 100% accuracy against real occlusion and other difficult scenarios. Table 2.4 also shows that among others, the RSR algorithm exhibits maximum robustness against viewpoint changes. It correctly recognizes all but one sequence which shows extreme viewpoint change i.e. when the direction of

Table 2.4: Results on the Robustness dataset: Performance under real occlusion, viewpoint changes and other difficult scenarios using LMP descriptors (trained on the Weizmann action dataset).

Performance under real occlusion and other difficult scenarios					
Test sequence	Gorelick et al. [39]	Shared-Hist	RSR	EFVC	Concat
walking with a dog	walk	walk	walk	walk	walk
swinging a bag	walk	walk	walk	walk	walk
walking in a skirt	walk	walk	walk	walk	walk
occluded legs	walk	walk	walk	skip	walk
occluded by a pole	walk	walk	walk	walk	walk
normal walk	walk	walk	walk	walk	walk
carrying briefcase	walk	walk	walk	walk	walk
knees up	walk	run	walk	skip	walk
limping walk	walk	walk	walk	walk	walk
sleepwalking	walk	walk	walk	walk	walk
Performance under viewpoint changes with the system only trained on subjects walking in 0°					
Test sequence	Gorelick et al. [39]	Shared-Hist	RSR	EFVC	Concat
walking in 0°	walk	walk	walk	walk	walk
walking in 9°	walk	walk	walk	walk	walk
walking in 18°	walk	walk	walk	walk	walk
walking in 27°	walk	walk	walk	walk	walk
walking in 36°	walk	walk	walk	walk	walk
walking in 45°	walk	walk	walk	walk	walk
walking in 54°	walk	walk	walk	walk	walk
walking in 63°	walk	skip	walk	side	walk
walking in 72°	walk	skip	walk	skip	skip
walking in 81°	walk	side	skip	side	side

walking in the test sequence is almost orthogonal to that in the training sequences. Recall that, the system is trained with the sequences from the Weizmann action dataset where the subjects are walking parallel to the camera i.e. in 0°.

The UCF sports dataset

The UCF Sports dataset [40] is considered to be one of the most challenging datasets in the field of action recognition. This dataset contains 149 action sequences collected from various sports videos which are typically featured on broad-

Figure 2.6: Sample frames from the UCF Sports dataset: diving (s1), golf swinging (s2), kicking (s3), lifting (s4), horse riding (s5), running (s6), skating (s7), swinging (s8) and walking (s9).

	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	**1**	0	0	0	0	0	0	0	0
s2	0	**.86**	0	0	0	0	0	0	.14
s3	0	0	**.48**	0	0	.24	.28	0	0
s4	0	0	0	**1**	0	0	0	0	0
s5	0	0	0	.14	**.43**	0	.14	0	.29
s6	0	0	0	0	.10	**.90**	0	**0**	0
s7	0	0	0	0	0	0	**.71**	0	.29
s8	0	0	0	0	0	0	0	**1**	0
s9	0	0	0	0	0	0	0	0	**1**

(a)

	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	**1**	0	0	0	0	0	0	0	0
s2	0	**1**	0	0	0	0	0	0	0
s3	.04	0	**.43**	0	0	.28	0	0	.25
s4	0	0	0	**1**	0	0	0	0	0
s5	0	.05	0	0	**.33**	.19	0	0	.43
s6	0	0	0	0	0	**1**	0	**0**	0
s7	0	0	0	0	0	.15	**.28**	0	.57
s8	0	0	0	0	0	0	0	**1**	0
s9	0	0	0	0	0	0	0	0	**1**

(b)

Figure 2.7: Results on the UCF sports dataset: (a) Cuboids + RSR (83.8%) and (b) Cuboids + concat (80.9%).

cast television channels such as BBC and ESPN. The collection represents a natural pool of actions featured in a wide range of scenes and viewpoints. The dataset also exhibits occlusion, cluttered background, variations in illumination, scale and motion discontinuity. The 9 actions are: diving, golf swinging, kicking, lifting, horse riding, running, skating, swinging and walking. Some of these sequences also contain more than one subjects.

The recognition results and confusion matrices are presented in Table 2.5 and Figure 2.7. The highest accuracy achieved in our experiments is 83.8% using cuboid features in combination with the RSR algorithm. Table 2.5 also compares our proposed algorithms with a number of existing ones. Apparently, our recogni-

Table 2.5: Comparison with state-of-the-art on the UCF Sports dataset.

Approach	Recognition accuracy (%)
Rodriguez et al. [40]	69.2
Yeffet & Wolf [51]	79.2
Zhu et al. [48]	84.3
Wang et al. [47]	85.6
Yao et al. [49]	**86.6**
Cuboids-based results	
Shared-hist	76.5
RSR	**83.8**
EFVC	82.8
Concat	80.9
LMP-based results	
Shared-hist	75.8
RSR	78.5
EFVC	77.1
Concat	77.8

tion rate is lower than those reported in some of the recent works [47–49]. However, unlike the methods such as [47, 48], we have *not* enlarged[1] the training set. We have also used a much smaller dictionary $[128 \times 512]$. The methods that use dense sampling with HOG3D descriptors [47, 48, 50] as features are computationally more demanding compared to the features we have used. The result reported in [49] is obtained using dense features, randomized trees and Hough transform-based voting. This method is also computationally more intense compared to our approach.

Both features and classifier contribute to the final recognition results. It is thus difficult to asses the contribution of our proposed classification methods by comparing it with methods that use different features. In order to find out the real

[1] in [47] and [48], the datasets are enlarged by adding horizontally flipped version of each video.

Table 2.6: Comparison of classification methods using the same features on the UCF Sports dataset.

Feature	Classification	Dictionary size	Accuracy (%)
cuboids-HOG	vector quant (shared) dict + non-linear SVM	4000	72.2
cuboids-HOG	sparse rep (shared) dict + linear SVM	512	**79.6**
cuboids-HOG	class dict + RSR	256	**83.8**

contribution of our approach, we concentrate on the results that are obtained using the same features. In Table 2.6, we compare our results with that the method [47] that use the same features as ours. Our method shows significant improvement in accuracy (more than 10%). This results also serve as a proof to that our sparse representation-based approach outperforms vector quantization-based methods in terms of accuracy and efficiency (note that, our method also uses smaller dictionaries).

2.3.3 Discussion

In order to perform any of the proposed classification algorithms, the two steps that require the bulk of computation are (i) the dictionary learning in the training stage and (ii) the sparse coding step in the classification stage. The dictionaries can be learnt offline as part of the training stage. The sparse coding however has to be performed during the classification which is of more importance to us. Our implementation uses an efficient sparse coding algorithm called the *Batch*-OMP [52]. Its computational complexity is $\mathcal{O}(mnk)$ per training signal, where the dictionary dimension is $m \times n$ and s is the sparsity constraint and $k << n$ [52].

To provide a practical idea of the run time, we provide the computation time for each of the classification algorithms proposed in this chapter for the same dataset, using the same training and test set. Shared-hist takes 0.03 sec, RSR takes 1.76 sec,

EFVC takes 2.30 sec and Concat takes 0.17 sec to classify the same test data.

We have shown that the class-specific dictionaries (or their concatenation) produce better recognition results compared to the shared dictionary. We advocate the use of class-specific dictionaries because along with superior results they also offer scopes to save computation. While Concat works very well on the relatively simpler Weizmann action database, RSR performs better on complex database like the UCF sports database. The success of RSR is due to its robustness to outliers.

We have employed RP as the fastest possible dimensionality reduction process. In this work the learnt dictionaries are overcomplete by a factor of 2 or 4. We have observed that for a given feature dimension, increasing the overcompleteness factor does not necessarily increase the accuracy; but it does raise the cost of computation significantly. From our experiments we found that increasing the overcompleteness factor beyond 4 does not improve the results much and in fact starts to fall for overcompleteness factors greater than or equal to 6.

LMP is introduced as a fast, light-weight spatio-temporal motion descriptor. It is interesting to notice that simple features like LMP can outperform sophisticated features like Cuboids in the case of Wiezmann database.

2.4 Summary

In this chapter, we have proposed four sparse representation-based classification algorithms: Shared-hist, RSR, EFVC and Concat. These algorithms have been shown to perform at par or better than the state-of-the-art for two important applications: image-based face recognition and video-based human action recognition. Another important observation made in this chapter is that the sparse modeling approach significantly outperforms (more than 10% improvement in accuracy) the traditional

vector-quantization based dictionary construction.

Chapter 3

Sparse Representation-based Image Similarity Measurement

Many image and multimedia information processing systems rely on the availability of a good image similarity measure. Despite the long history of image similarity evaluation, open issues still exist. These include the need of developing generic similarity measures which do not assume any prior knowledge of the application or the data. In this chapter, we develop such a generic method for measuring image similarity based on learning sparse representations. The proposed method encodes the information content of one image using the information from the other image, and use the sparsity of the representation as a measure of similarity between the two images.

3.1 Background and motivation

Measuring the similarity between a pair of images is of critical importance to many image processing systems involving retrieval, enhancement, copy detection, qual-

ity assessment, clustering and classification. Given the long history of image similarity evaluation, the volume of literature on this topic is large and diverse.

Widely used similarity measures such as the Euclidean distance, the Mean Squared Error and other norm-based measures work well in specific cases, but they are often criticized for not corresponding well with our visual perception of similarity [53]. Another popular approach involves describing the visual content of images by extracting a set of meaningful features. The similarity between two images is then computed in terms of the similarity between their features. However, the success of this approach is limited by the availability, selection and extraction of a good set of meaningful features and these demand specific knowledge of the application and of the data.

Recently, there has been an interest in developing image similarity measures using *compression* methods [54–58]. In this new line of research, two signals are considered similar if one can be compressed significantly when the information of the other is provided. The advantages of these methods are that they are *parameter-free* (the only choice the user has to make is which compression algorithm to use), and *generic* (they assume no prior knowledge of the application, and can be applied, without modification, to a variety of problems).

3.1.1 Compression-based similarity methods

The compression-based similarity methods rely on a new mathematical theory of similarity which is in turn based on the idea of the *Kolmogorov complexity* [54, 55]. The work of Kolmogorov and others [59–61] on how to measure data complexity has been influential in many areas of knowledge, across multiple disciplines. The notion of complexity of a string is related to its randomness. For example,

the binary string 1101010001 is considered more complex compared to the string 0101010101, because the latter contains a regularity (repeating pattern) and therefore is less random. Kolmogorov complexity formalizes this concept:
Given a finite object, such as a binary string Y, its Kolmogorov complexity $K(Y)$ is defined as the length of the shortest program that can effectively produce Y on a universal computer, such as a Turing machine [62].

The Kolmogorov complexity (also known as the *algorithmic entropy*) is however a non-computable quantity in general. In practice, it is often approximated by the *length or the file size of the compressed data*. Intuitively, the more a given data can be compressed, the lower is its complexity and vice versa. Recently, Kolmogorv's theory of complexity has been used to address the problem of similarity measurement. Given two signals Y and Z, a distance metric, known as the *Normalized Information Distance* (NID) is developed using $K(Y)$ and the conditional Kolmogorov complexity $K(Y|Z)$ [54, 55].

$$\mathrm{NID}(Y,Z) = \frac{\max\{K(Y|Z), K(Z|Y)\}}{\max\{K(Y), K(Z)\}} \tag{3.1}$$

where the *conditional Kolmogorov complexity* $K(Y|Z)$ is defined as the length of the shortest program used by a universal computer to generate Y when Z is known.

Due to the non-computable nature of the Kolmogorov complexity, a practical analog of the NID metric (defined in Equation 3.1) is proposed based on standard compression methods. This is called the *Normalized Compression Distance* (NCD). Intuitively, NCD considers Y and Z to be similar if one can be significantly

compressed when the information of the other is provided. It is defined as follows:

$$\mathrm{NCD}(Y,Z) = \frac{\max\{C(Y|Z), C(Z|Y)\}}{\max\{C(Y), C(Z)\}} \tag{3.2}$$

where $C(X)$ is the length or size of the compressed version of X. The *conditional compression* $C(Y|Z)$ is approximated as follows:

$$C(Y|Z) = C(YZ) - C(Z) \tag{3.3}$$

where $C(YZ)$ denotes the compressed length of the concatenation of Y and Z.

The NCD metric has been shown to be effective in clustering mitochondrial genomes, languages and music [55]. Following the success of NCD, different versions of compression-based distance measures have been proposed; for example, a *Compression-based Dissimilarity Measure* (CDM) is proposed in the context of parameter-free data mining and is shown to be useful for anomaly detection, clustering and classification of text, DNA and time-series data [63]. CDM is defined as

$$\mathrm{CDM}(Y,Z) = \frac{C(YZ)}{C(Y)+C(Z)} \tag{3.4}$$

Other applications of compression-based distances include symbolic music clustering [64] and plagiarism detection [65]. The idea of compression, independent from NCD, has also been used to design a pattern representation scheme for automatic categorization of music, voice, genome, etc. [56]; but this method requires encoding media data input into text.

3.1.2 Compression-based image similarity

The compression-based similarity measures have been shown to be highly effective in clustering and classifying discrete, uni-dimensional data such as text and protein sequences [54, 55]; but their the successful application in the context of real-valued higher dimensional data such as images has been scarce. We identify two major reasons behind that.

- The success of the compression-based distances heavily depends on the availability of a *normal* compressor. A compressor is said to be normal only if it satisfies certain conditions such as idempotency, monotonicity, symmetry, etc. (please refer to [55] for details). The problem is that most state-of-the-art image compressors (such as JPEG, JPEG2000) are *not* normal, and normal compressors (such as the compressors of the Lempel-Ziv family) do not work well on images [57].

- Another serious obstacle lies in evaluating and approximating the conditional complexity terms such as $C(Y|Z)$ in NCD in Equation 3.2. These conditional terms are the key components in a compression-based measure. The existing compression-based methods (whether or not they involve images) either approximate the conditional compression $C(Y|Z)$ by $C(YZ) - C(Z)$ or use a simplified definition so as not to include any conditional term (as in Equation 3.4). Direct evaluation of $C(Y|Z)$ is usually bypassed mainly to retain the simplicity of the compression-based measures since evaluating $C(Y|Z)$ accurately requires delving into the complicated standards and algorithms of data or image compression. This also makes the compression-based methods difficult to improve upon.

Clearly, the straightforward extension of the methods that work well on discrete, one-dimensional data has not been very promising in the context of images. In the pursuit of alternatives, a new image encoder has been proposed based on the finite context model and preliminary results on a face database are provided [57]. Another recent approach, namely the CK-1 method, uses the MPEG1 video compressor to measure image similarity [58]. This method takes advantage of the temporal redundancy reduction step in video compression which performs inter-frame block matching. In this approach, a two-frame video consisting of the images to be compared is created. One frame is compressed with reference to the other frame using a standard video compressor. The compressed file size of the video is used to approximate the closeness between the pair of images. This method has been shown to be useful in texture classification.

3.2 Proposed approach

A natural way of measuring the similarity between two given images is to quantify how well each image can be represented using the information of the other. The more similar the images, the better is the representation of one image in terms of the other. Our method formalizes this intuitive idea of similarity using a sparse representation-based approach. Given a pair of images, our method learns a dictionary for each image and computes how sparsely can one image be approximated using the dictionary extracted from the other, with a required precision.

3.2.1 Sparsity as a measure of data complexity

It is well-known that sparsity of representation plays a key role in achieving good compression. For example, the superiority of JPEG2000 is mainly attributed to the

capability of the wavelet transform toward representing an image more sparsely than the DCT used in JPEG. Intuitively, the more sparse the representation of a signal is, the fewer are the components needed to capture the signal's information content and the better it can be compressed.

Sparsity thus can be seen as a direct measure of the randomness or complexity of the data. A natural image usually exhibits many repeated structures which can be discovered through its decomposition over a set of properly chosen basis functions. Due to the presence of redundancy, only a few basis functions are required to capture the significant information content of such images, resulting in a sparse representation. In the case where such structures are rare (e.g. in random Gaussian noise), there is no way to represent the data using a small number of basis elements. This indicates that as the complexity of a signal increases, more and more components are needed to represent the signal with a desired accuracy i.e. its sparsity decreases in the transform domain. This inherent connection between sparsity and data complexity is exploited in our proposed distance measure.

3.2.2 Sparse representation-based distance measure

Let us consider an image $Y \in \mathbb{R}^N$. A set of s random, possibly overlapping patches (each of dimension $\sqrt{n} \times \sqrt{n}$) is extracted from Y. Every patch is converted to a vector of length n and the patches are concatenated to form a matrix $\mathbf{B}_Y \in \mathbb{R}^{n \times s}$. In order to build a perceptually meaningful model for Y, we intend to learn an overcomplete dictionary $\Phi_Y \in \mathbb{R}^{n \times m}$ that has n atoms ($n < m$) using the local patches in $\mathbf{B}_Y$ as input. However, greater difficulties arise with a set of overcomplete bases. An overcomplete dictionary matrix creates an underdetermined system of linear equations having an infinite number of solutions. Knowing that the natural signals

are sparsely representable, often in such cases, we seek the sparsest solution.

Our objective is to learn Φ_Y such that each patch (column) $\mathbf{b}_{Y_i} \in \mathbf{B}_Y$ can be closely approximated as a linear superposition of a small number of atoms in Φ_Y. This is achieved by solving the following sparse optimization problem:

$$\min_{\{\Phi_Y, \mathbf{x}_Y\}} \quad \sum_i \|\mathbf{x}_{Y_i}\|_p \quad \text{s.t.} \quad \forall i, \|\mathbf{b}_{Y_i} - \Phi_Y \mathbf{x}_{Y_i}\|_2 \leq \delta \tag{3.5}$$

where the vector $\mathbf{x}_{Y_i} \in \mathbb{R}^m$ is the sparse representation of the patch $\mathbf{b}_{Y_i} \in \mathbb{R}^n$. The sparse representation of $\mathbf{B}_Y$ w.r.t. Φ_Y is denoted as the matrix $\mathbf{X}_Y = [\mathbf{x}_{Y_1}|\mathbf{x}_{Y_2}|...|\mathbf{x}_{Y_s}]$. The value of p in the ℓ_p norm in Equation 3.5 is typically 0 or 1, and δ denotes the reconstruction error controlled by the user. For the ℓ_0 case, we employ the K-SVD algorithm [23] which provides a greedy approximate solution to Equation 3.5 .

Sparse representation-based complexity functions

We define two quantities that measure the compressibility of an image i.e. how much can an image be compressed. These two quantities use (i) the dictionary learnt from the image itself, and (ii) the dictionary extracted from the other image, Z. We name these terms as the *Sparse complexity* and the *Relative sparse complexity*, respectively.

Definition 1. *Given an image Y, its Sparse Complexity $S_\delta(Y, \Phi_Y)$ is defined as the sparsity of $\mathbf{X}_Y$ averaged over the number of columns in $\mathbf{X}_Y$ i.e.*

$$S_\delta(Y, \Phi_Y) = \frac{1}{s} \|\mathbf{X}_Y\|_p = \frac{1}{s} \sum_{i=1}^{s} \|\mathbf{x}_{Y_i}\|_p \tag{3.6}$$

Therefore, for $p = 0$, $S_\delta(Y, \Phi_Y)$ is the average number of non-zero coefficients

required to reconstruct a column of $\mathbf{B}_Y$ using Φ_Y, up to a required precision δ. Smaller value of $S_\delta(Y,\Phi_Y)$ indicates higher compressibility (i.e. lower complexity) of Y.

Properties of $S_\delta(Y,\Phi_Y)$:

- $S_\delta(Y,\Phi_Y) > 0$ for non-empty Y, and is equal to 0 otherwise.
- Considering that Y is represented by $\mathbf{X}_Y$ and hence YY is represented by $[\mathbf{X}_Y|\mathbf{X}_Y]$, we have $S_\delta(YY,\Phi_Y) = S_\delta(Y,\Phi_Y)$. This property (idempotency) follows from the averaging operation and indicates that the sparse complexity function can compress the duplicate entries.

Given another image Z, the compression-based measures attempts to approximate how much can the image Y be compressed when information about Z is available. As discussed before, this conditional quantity is difficult to approximate and this limits the success of these measures. We hence define a slightly different complexity term that measures *how much information about Y is contained in Z*. We name this term as the *Relative Sparse Complexity*, .

Let $\Phi_Z \in \mathbb{R}^{n\times m}$ be the dictionary pertaining to the image Z and is learnt in the same manner as Φ_Y (refer to Equation 3.5). The image Y can be approximated in terms of the dictionary of Z as follows:

$$\min_{\mathbf{x}_{Y|Z}} \quad \sum_{i=1}^{s} \left\|\mathbf{x}_{Y|Z_i}\right\|_p \quad \text{s.t.} \left\|\mathbf{b}_{Y_i} - \Phi_Z \mathbf{x}_{Y|Z_i}\right\|_2 \leq \delta \tag{3.7}$$

where $\mathbf{x}_{Y|Z_i} \in \mathbb{R}^m$ is the sparse representation of $\mathbf{b}_{Y_i}$ w.r.t. Φ_Y.

$\mathbf{X}_{Y|Z} = [\mathbf{x}_{Y|Z_1}|\mathbf{x}_{Y|Z_2}|...|\mathbf{x}_{Y|Z_s}]$ is the sparse representation of $\mathbf{B}_Y$ w.r.t. Φ_Y.

Definition 2. *Given two images Y and Z, the Relative Sparse Complexity $S_\delta(Y,\Phi_Z)$ is defined as the sparsity of $\mathbf{X}_{Y|Z}$ averaged over the number of columns in $\mathbf{X}_{Y|Z}$.*

$$S_\delta(Y,\Phi_Z) = \frac{1}{s}\left\|\mathbf{X}_{Y|Z}\right\|_p = \frac{1}{s}\sum_{i=1}^{s}\left\|\mathbf{x}_{Y|Z_i}\right\|_p \tag{3.8}$$

Therefore, for $p = 0$, $S_\delta(Y,\Phi_Z)$ becomes the average number of non-zero coefficients required to reconstruct a column of $\mathbf{B}_Y$ using Φ_Z, up to a required precision δ. A smaller value of $S_\delta(Y,\Phi_Z)$ indicates that Y is efficiently represented by the information extracted from Z i.e. Y and Z have higher similarity.

Properties of $S_\delta(Y,\Phi_Z)$:

- $S_\delta(Y,\Phi_Z) > 0$ for non-empty Z, and 0 otherwise.
- $S_\delta(YZ,\Phi_Z) = S_\delta(ZY,\Phi_Y)$ (symmetry)
- $S_\delta(Y,\Phi_Z) > S_\delta(Y,\Phi_Y)$ for $Y \neq Z$. This is because, in general, Y is expected to be more efficiently (sparsely) approximated using Φ_Y - the dictionary trained on itself, than Φ_Y - a dictionary trained on a different image.

The distance measure

Based on the two terms defined above, a sparse representation-based distance measure $\mathscr{D}$ is defined as follows:

$$\mathscr{D}(Y,Z) = \frac{S_\delta(Y,\Phi_Z) + S_\delta(Z,\Phi_Y)}{S_\delta(Y,\Phi_Y) + S_\delta(Z,\Phi_Z)} - 1 \tag{3.9}$$

The proposed form of $\mathscr{D}$ is much similar to that of the compression-based CK-1 distance measure [58]. From the property of the relative sparse complexity we have

$$S_\delta(Y,\Phi_Z) > S_\delta(Y,\Phi_X) \text{ and } S_\delta(Z,\Phi_Y) > S_\delta(Z,\Phi_Y)$$

Hence,

$$\frac{S_\delta(Y,\Phi_Z)+S_\delta(Z,\Phi_Y)}{S_\delta(Y,\Phi_Y)+S_\delta(Z,\Phi_Z)} > 1 \text{ for } Y \neq Z.$$

Intuitively, $\mathscr{D}$ measures how efficient, on average, is it to approximate one image Y using the information of Z extracted in the form of a dictionary of its dominant local structures. The smaller the values of $\mathscr{D}$ the higher is similarity between the two images.

Properties of $\mathscr{D}$:

- *Non-negativity:* $\mathscr{D}$ is always non-negative, the lowest value of $\mathscr{D}$ is 0 when $Y = Z$.

- *Symmetry:* Clearly, $\mathscr{D}$ is symmetric i.e. $\mathscr{D}(Y,Z) = \mathscr{D}(Z,Y)$. Symmetry is an important property for a similarity or dissimilarity measure because many algorithms (e.g. spectral clustering) rely on this property.

- *Metricity:* $\mathscr{D}$ does not follow the metric axiom of triangle inequality and hence cannot be called a metric. It would have been mathematically convenient if $\mathscr{D}$ was a metric. However, many researchers have argued that perceptual distances are typically non-metric in nature [66, 67].

Note that, we have used $p = 0$ to compute the complexity functions because

our dictionary learning method uses greedy ℓ_0 approximation. If ℓ_1 optimization is used to learn the dictionaries, it would be better to use $p = 1$ for the definitions.

3.3 Performance evaluation

In order to establish the generality of the proposed distance measure, we perform experiments on a variety of applications. We first carry out experiments to evaluate the compatibility of the proposed measure with the human perception of similarity. This is followed by clustering, retrieval and classification experiments involving larger datasets. The datasets that we choose contain real-world images from different domains like biology, biometrics, medicine and natural textures.

3.3.1 Implementation details

Practically, there are 4 parameters to be set: the patch size ($\sqrt{n}$), the number of patches to be extracted from each image (s), the number of dictionary elements (m) and the reconstruction error (δ). Unfortunately, there is no theoretical guidelines to determine the values of these parameter, so we rely on previous work and empirical methods. We have used the same parameter values for all experiments, unless mentioned otherwise. Below, we describe how the parameter values are chosen for this particular work.

Patch size ($\sqrt{n}$) and automatic scale selection: The patch size determines the spatial scale at which an image is analyzed. For simplicity and speed, we analyze each image at a single scale, but use a simple technique to *automatically* select the (sub)optimal scale. A 2D LOG filter is applied to each image to detect the local maxima points (keypoitns) at four different scales. The scale at which the maximum number of keypoints are detected is chosen as the (sub)optimal scale

for that image. The image is downsampled accordingly and a set of patches are extracted. For example, if the scale is found to be 2, the image is downsampled by a factor of 2 and then patches of size 8×8 i.e. $\sqrt{n} = 8$ are extracted. This particular patch size is chosen in order to be consistent with most of the compression based algorithms (e.g. JPEG1) which process 8×8 blocks. The automatic scale selection is performed on all images for all datasets except for the VVT Wood dataset due to the small dimensions (64×64) of the original images.

Number of patches (s): In order to train a dictionary, a large number of patches need to be extracted. The color images are first converted to grayscale to achieve *color invariance*. It is also important that the randomly extracted patches contain important structural information of the image and do not come from the homogeneous regions of the image only. This is accomplished by selecting the patches whose energy levels are above an empirically set threshold. A collection of $s = 3000$ such patches are extracted from every image and is used to train its corresponding dictionary. The input patches for dictionary learning have zero mean and unit standard deviation which account for *luminance and contrast invariance*.

Overcompleteness (m/n): Since we intend to learn an overcomplete dictionary, we must have $m > n$. The ratio m/n is called the *overcompleteness factor*. It has been shown that for small overcompleteness factor, sparse representation is stable in the presence of noise [68]. Thus we set $m/n = 2$, where $n = 64$.

Reconstruction error (ε): We used $\delta = 0.1$ which means that the input vector is reconstructed with at least 90% accuracy. Note that a lower reconstruction error can produce a better dictionary, but requires more computation and more importantly, may cause overfitting.

(a) originial image
PSNR $= \infty$, VIF $= 1$
Proposed distance $= 0$

(b) contrast change
PSNR $= 24.53$, VIF $= 1.50$
Proposed distance $= 0.17$

(c) luminance change
PSNR $= 15.97$, VIF $= 0.95$
Proposed distance $= 0.20$

(d) white noise
PSNR $= 31.95$, VIF $= 0.96$
Proposed distance $= 0.33$

(e) lossy jpeg
PSNR $= 28.47$, VIF $= 0.92$
Proposed distance $= 0.38$

(f) unrelated image
PSNR $= 13.21$, VIF $= 0.14$
Proposed distance $= 0.54$

Figure 3.1: Comparison of the proposed distance measure with human perception and the well-known perceptual similarity method VIF.

3.3.2 Correlation with human perception

It is important that the distance measure between images correlate with human perception. We begin with measuring the similarities between a reference image (Figure 3.1(a)) and its distorted versions (Figure 3.1(b)-(e)) as well as a completely unrelated image (Figure 3.1(f)). We also compare our results with PSNR and the well-known Visual Information Fidelity (VIF) similarity measure [69] (values closer to zero indicates lower similarity). Figure 3.1 shows that our proposed distance measure $\mathscr{D}$, PSNR and VIF correlate well with human perception.

Next, we perform a simple clustering task where it is possible to evaluate the

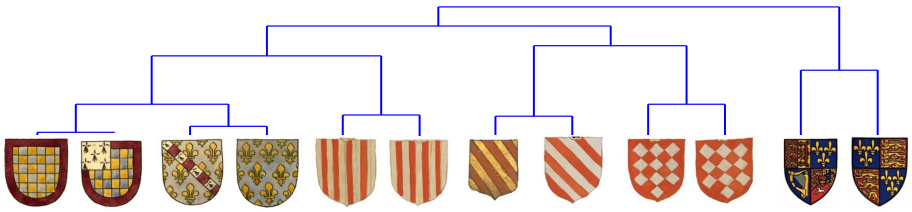


Figure 3.2: Hierarchical clustering result on the Heraldic Shields dataset using the proposed sparse representation-based distance measure (although color images are shown here the result is obtained using grayscale images).

results manually. The *Heraldic Shields dataset* [58] (see Figure 3.2) contains 12 images (of various sizes) which are to be clustered into 6 pairs. All possible pairwise distances are computed using the proposed distance measure $\mathscr{D}$. Hierarchical clustering is performed using the average linkage method. The clustering result shown in Figure 3.2 demonstrates that our measure has discovered all 6 basic pairs of shields, and corresponds well with human intuition.

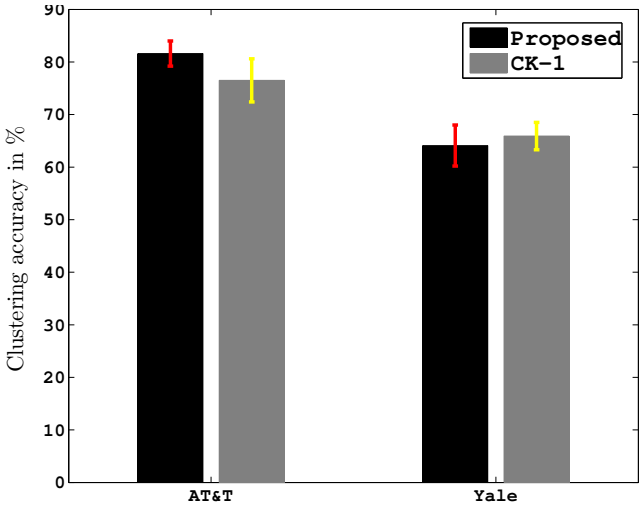
3.3.3 Clustering facial images

In this segment, we move towards more difficult clustering problems involving two larger benchmark datasets:

AT&T face [70]: This dataset contains 400 facial images of 40 individuals in 10 poses. These images (dimension: 112×92) are taken at different times with varying illumination, facial expressions and details.

Yale face [71]: This dataset has 165 grayscale facial images of 15 individuals. There are 11 images per subject, one per different condition: center light, with glasses, happy, left light, no glasses, normal, right light, sad, sleepy, surprised, and wink.

(a)



(b)

Figure 3.3: (a) Sample images from the AT&T (first 3) and the Yale face (last 3) databases; (b) Clustering accuracy for the AT&T face (Proposed: $81.6 \pm 2.4\%$, CK-1: $76.5 \pm 4.1\%$) and the Yale face (Proposed: $64.1 \pm 3.9\%$, CK-1: $65.9 \pm 2.6\%$) databases.

For each dataset, an $M \times M$ similarity matrix is computed using (3.9), where M is the number of elements in the dataset. This similarity matrix serves as the input to a standard spectral clustering algorithm [72]. The accuracy of the clustering results is measured using the Hungarian algorithm [73]. We compare our results with the compression-based state-of-the-art CK-1 distance measure [58] using the code provided by the authors. Due to the initialization process in spectral clustering, the accuracy varies slightly at each run. Figure 3.3 reports the mean clustering accuracies along with the standard deviations as computed over 10 runs for the two databases under consideration. The proposed measure outperforms CK-1 on the

AT&T face dataset by 5.1% and its performance is 1.8% lower than CK-1 on the Yale dataset. We also performed a hypothesis test for both datasets. While the performance improvement for the AT&T dataset was found significant, our different in accuracy on the Yale dataset is not statistically significant.

3.3.4 Texture retrieval

An image retrieval system, when provided with a query image, returns images from a large dataset that are perceptually similar to the query. We perform standard retrieval experiments on the following benchmark texture dataset.

Brodatz texture dataset [74]: This is a benchmark dataset that contains a variety of natural textures like grass and cloth (see Figure 3.5). There are 111 different texture classes. Each original texture image is divided into 9 subimages to create the samples for that class.

For each query, the distances between the query and the remaining 998 images in the dataset are computed, and the first K nearest images are retrieved. The performance of a retrieval system is often measured in terms *Precision* and *Recall accuracy*. Precision is defined as the ratio of correctly retrieved images to the total number of images retrieved. Recall accuracy is defined as the ratio of the number of correctly retrieved images to the number of images available for the query class. Both precision and recall accuracy are expressed in terms of %. Our retrieval results are compared with those obtained using the CK-1 method in Figure 3.4 where our method clearly outperforms CK-1.

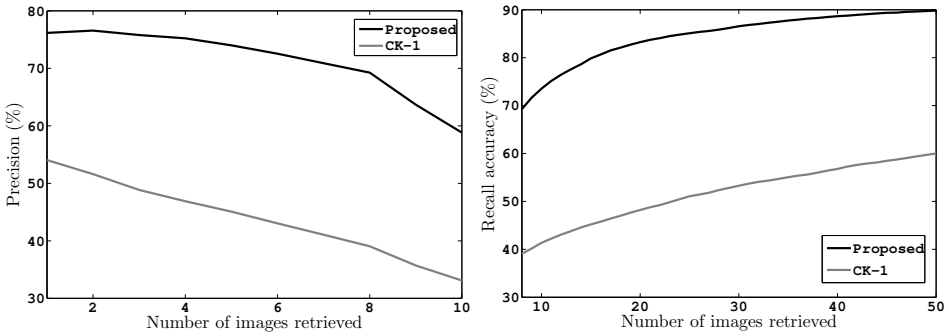


Figure 3.4: Shown are the image retrieval results in terms of precision (left) and recall accuracy (right) obtained using the proposed method and the compression-based state-of-the-art CK-1 method on the Brodatz dataset.

3.3.5 Classification

Supervised classification experiments are performed on a diverse collection of image datasets drawn from the sources across various disciplines such as biology, medicine, forensics, etc. Sample images from each dataset are presented in Figure 3.5 and a brief description of each dataset is provided below:

UIUCTex [75]: This dataset features 25 texture classes with 40 samples each.

KTH Tips [76]: This dataset consists of textures of 10 different materials. The images vary in illumination, pose and scale.

Camouflage [58]: This dataset consists of 80 images of 9 varieties of modern US military camouflage. The images are created by photographing military t-shirts at random orientations.

Nematodes [58]: Nematodes are wormlike animals with great commercial and medical importance. Their species are often very difficult to distinguish from each other. This dataset contains 50 images of 5 different species of nematodes.

Tire tracks [58]: This is a collection of tire imprints left on a paper. It has 48

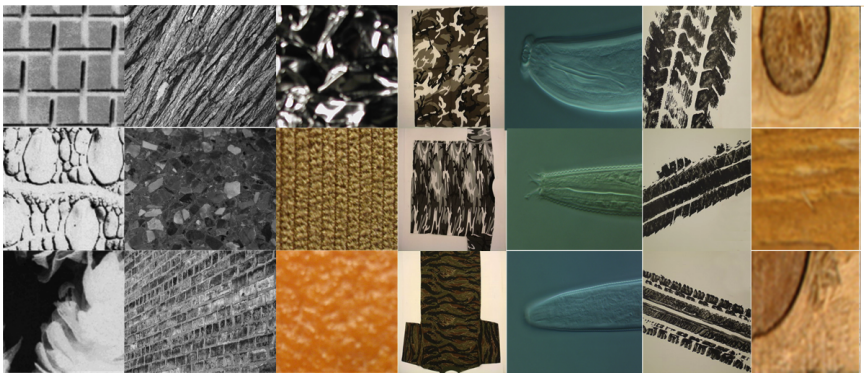


Figure 3.5: Sample images from the various datasets: (column wise, from left) Brodatz,UIUC, KTH, Camouflage, Nematode, Tire tracks, and Woods.

imprints of 3 different tires at varying directions.

VVT Wood [58]: This dataset contains 200 images of 40 types of wood defects (such as dry knot and small knot, etc.). The task is to label an image as either defective or sound.

The classification results for the above datasets using the proposed method and the CK-1 are presented in Table 3.1. We test both methods using a leave-one-out scheme in a 1-NN framework. Our method demonstrates much better or comparable accuracy for all the datasets.

3.3.6 Discussion

Most compression-based methods use an off-the-shelf compressor (data, image or video compressor) and treat the compressor as a black-box. This makes it difficult to understand which part of the compression algorithm actually estimates the complexity of the data or measures the similarity. Consequently, the compression-based methods are difficult to improve upon, unless one wants to delve into the

Table 3.1: Classification accuracy on various datasets obtained using the proposed distance measure and the state-of-the-art compression-based distance CK-1.

Dataset	Classes	Proposed (%)	CK-1 [58] (%)
Brodatz	111	**76.2**	54.0
UIUCTex	25	**51.6**	51.0
KTH Tips	10	84.5	**86.0**
Camouflage	9	87.0	**87.5**
Nematodes	5	**62.0**	56.0
Tire tracks	3	**79.2**	**79.2**
VTT wood	2	**85.2**	80.5

details of the compression algorithms.

The proposed method takes a rather direct approach towards the approximation of complexity, and it is easier to understand and improve. Our method can be easily extended to measure the similarity between any type of signals including audio, video and other type of images such as medical images.

The proposed method requires learning a dictionary for each image. The dictionary learning process takes only a few seconds; for example, with the above-mentioned parameter values, a MATLAB implementation takes ~ 2 secs to learn a dictionary per image (including the patch extraction process) on a standard PC (intel quad @2.67GHz). This is as fast as any standard feature extraction process. However, our method is still slower compared to the compression-based CK1 measure. This can be explained by the fact that the areas of dictionary learning and sparse representation are still in the developing stage. In other words, unlike the standard compression algorithms, the existing algorithms for learning dictionaries or sparse representations are not yet fully optimized for speed or memory.

We have used a greedy algorithm (OMP) to solve the sparse optimization prob-

lems in this work, primarily for speed and simplicity. Better results may be achieved using ℓ_1 regularized algorithms but at a higher computational cost. The proposed method is also not parameter-free, it requires a few parameters to be set by the user.

3.4 Summary

In this chapter, we developed a generic measure of similarity between two images. Two images are considered similar if one can be compressed significantly when the information of the other is known. Given a pair of images, X and Y, our proposed method encodes the information content of X using the information from Y and vice versa. The compactness (sparsity) of the representation of X w.r.t. the information from Y is used as a measure of compressibility of X i.e. how much X can be compressed. The more sparse the representation of an image, the better it can be compressed and the more it is similar to the other image. The efficacy of the proposed measure is demonstrated through the high accuracies achieved in image clustering, retrieval and classification.

Chapter 4

Sparse Representation-based Perceptual Image Quality Assessment

Image quality assessment can be considered as a special case of image similarity measurement. A highly promising approach to assess the quality of an image involves comparing the structural information in this image with that in its reference image. The extraction of the structural information that is perceptually important to our visual system is however a challenging task. In this chapter, we develop a sparse representation-based approach to address this issue and propose a new image quality assessment metric called the Sparse Representation-based Quality (SPARQ) index.

4.1 Background and motivation

Digital images incur a variety of distortions during their acquisition, compression, transmission, storage or reconstruction. Such processes often degrade the visual quality of images. In order to monitor, control and improve the quality of images produced at the various stages, it is important to *automatically* quantify the image quality. Since the end-users of the majority of image-based applications are humans, this requires the understanding of human perception of image quality, to be able to mimic it as closely as possible.

The Mean Squared Error (MSE) and the Peak Signal to Noise Ratio (PSNR) have been traditionally used to measure the image quality degradations. These metrics are mathematically convenient to use but they do not correlate well with human perception of image quality [53]. A considerable amount of research effort has been put towards quantifying the quality of images as *perceived* by humans, and a number of *objective* image quality assessment algorithms that agree with the subjective judgment of human beings have been developed. The objective quality assessment methods, depending on how much information about the original undistorted image they use, are broadly classified into three categories: *no-reference*, *reduced-reference* and *full-reference*. This work concentrates on the *full-reference* quality estimation approach.

The earlier focus of full-reference image quality assessment research has been on building a comprehensive and accurate model of the HVS and its psychophysical properties, such as the contrast sensitivity function. In this approach, the errors between the distorted and the reference images are quantized and pooled according to the HVS properties [77]. These methods require precise knowledge of the view-

ing conditions and are computationally demanding. Despite this complexity, the HVS *modeling-based* methods can only make linear or quasilinear approximations of the highly non-linear HVS. Our current understanding of the HVS is also limited in many aspects. Consequently, these methods do not yield highly superior results than that produced by MSE or PSNR MSE or PSNR [78].

The interest in modern image quality estimation research has therefore shifted to modeling the visual content of images based on certain significant properties of the HVS. This *visual fidelity-based* approach is more attractive because of its practicality and mathematical foundation [79, 80]. The majority of these fidelity-based methods attempt to quantify the perceptual quality either in terms of *statistical information* [81, 69] or in terms of *structural information* of the images [78, 82–86]. The statistical approaches hypothesize that the HVS has evolved over the years to extract information from natural scenes and therefore, use natural scene statistics to estimate the perceptual quality of images. The structural approaches on the other hand operate on the basis of a rather important aspect of the HVS - its sensitivity towards the image structures for developing cognitive understanding. In this approach, image quality is estimated in terms of the *fidelity of structures* between the reference and the distorted images.

The image quality metric that is representative of the class of structural information-based metrics is the Structural Similarity Index (SSIM) [82]. SSIM treats the non-structural distortions (such as, luminance and contrast change) separately from the structural distortions. The quality of a patch in the distorted image is measured by comparing it with the corresponding patch in the original image in terms of three components: luminance, contrast and structure. A global quality score is computed by combining the effects of the three components over all image patches.

SSIM achieved much success because of its simplicity, and its ability to tackle a wide variety of distortions. Due to its pixel-domain implementation, SSIM is highly sensitive to geometric distortions like scaling, translation, rotation and other misalignments [77]. To improve the performance of SSIM, multiscale extension [83], wavelet transform-based modification [86], gradient-domain implementation [84] and various pooling strategies [85, 87] have been proposed.

The underlying assumption behind utilizing the structural information is that the HVS uses the structures extracted from the viewing field for its cognitive understanding. Therefore, for an image to be considered of high-quality, all the structural information present in its reference image should be well preserved. From this viewpoint, the efficient capture of the structural information of images is the key to developing a successful image quality assessment algorithm. But extracting or analyzing the structural information in a perceptually meaningful way is a non-trivial task. A widely used mathematical tool for analyzing image structures is the wavelet transform. Its basis elements, being spatially localized, oriented and of bandpass in nature, resemble the receptive field of simple cells in the mammalian primary visual cortex (also known as V1 or the striate cortex) [4, 77]. The wavelet transform however uses a set of predefined, data-independent basis functions. Therefore its success is often limited by the degree as to how suitable the basis functions are in capturing the structure of the signals under consideration.

We propose the use of a more generalized approach to analyzing image structures in the context of image quality assessment. This involves *learning* from the training data a set of basis elements that could be adapted to represent the inherent structures of the signal in question. These learnt basis elements are collectively known as a *dictionary*. As each basis vector could be tailored to represent a sig-

nificant part of the structures present in the given data, a learnt dictionary is more efficient in capturing the structural information compared to a predefined sets of bases. In the last few years, several practical dictionary learning algorithms have been developed [23, 21]. It has been shown that the data-dependent, learnt dictionaries, due to their superior ability to efficiently model the inherent structures in the data, can outperform predefined dictionaries like wavelets in several image processing tasks [23, 2, 26]. More importantly, as mentioned in Chapter 1, this approach empowers us to build a *cortex-like* representation of an image.

In this chapter, we develop a full-reference image quality assessment metric which we call the Sparse Representation-based Quality (SPARQ) *index*. This metric relies on capturing the inherent structures of the reference image as a set of basis vectors which collectively form an overcomplete dictionary. These vectors are designed such that any structure (patch) in the image can have a sparse representation w.r.t. the dictionary. To estimate the visual quality of the distorted image the structures (patches) in this image are compared with those in the reference image, in terms of the learnt dictionary. Since our method analyzes image structures by building a cortex-like model of the stimuli, we expect the extracted structural information to be important to the HVS, and perceptually more meaningful compared to the structural information used in existing methods.

To evaluate the efficacy of the proposed metric, we perform various experiments on six publicly available, subject-rated image quality assessment datasets: A57 [88], CSIQ [89], LIVE [90], MICT [91], TID [92] and WIQ [93]. The proposed SPARQ index consistently exhibits high correlation with the subjective scores and often outperforms its competitors.

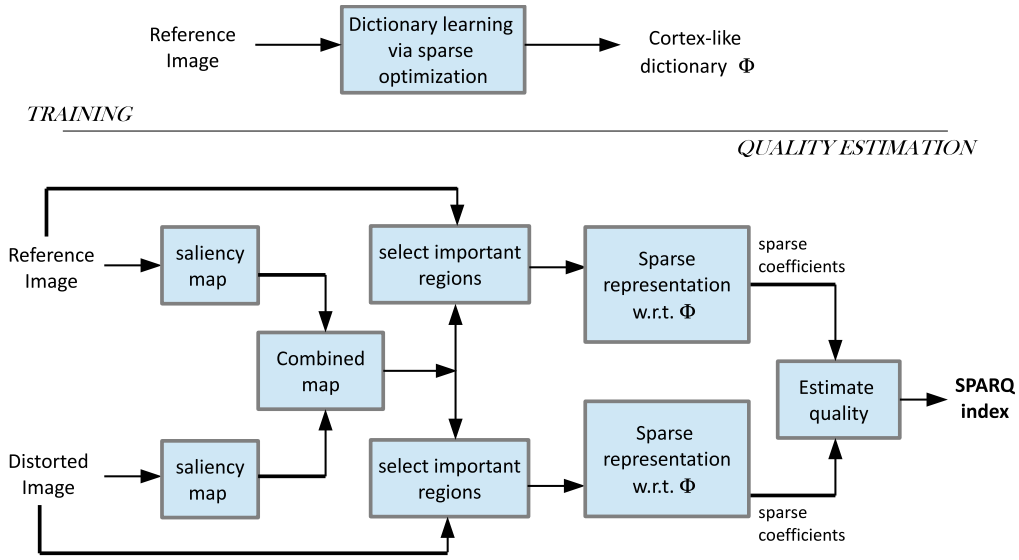


Figure 4.1: Overview of the proposed image quality assessment approach

4.2 Proposed approach

Our image quality assessment approach is divided into two phases:

- *training* phase - captures the inherent structures from the reference image by learning an overcomplete dictionary.
- *quality estimation* phase - generates a quality score for a given distorted image by comparing the structures in this image with the corresponding ones in its reference image, in terms of the learnt dictionary.

Figure 4.1 presents an overview of the proposed approach and each step is described below in detail.

4.2.1 Training phase

The motivation of this step comes from the very process of image formation and how an image is perceived by the HVS. The natural viewing field is highly structured and spatially correlated. The light rays that reflect off various structures in the viewing field, get focused onto an array of photoreceptors present in the retina. The visual information is then encoded in the form of complex statistical dependencies among the photoreceptor activities [94]. The goal of the primary visual cortex, as indicated in several seminal studies [4, 94], is to reduce these statistical dependencies in order to discover the intrinsic structures that gave rise to the image.

A reasonable strategy towards mimicking this phenomena is to describe the image in terms of a linear superposition of a small number of basis vectors. These basis vectors form a subset of a larger, overcomplete set of basis vectors (dictionary) that are adapted to the given image so as to best represent all structures in that image [4, 94]. It has been shown that on employment of this strategy, the resulting basis elements of the dictionary are qualitatively similar to the receptive field of the cortical simple cells [4]. The importance of sparsity as an important prior, as shown in [4], the sparsity is based on the observation that natural images contain sparse structures and can be described by a small number of structural primitives like lines, edges and corners [94, 95]. Due to overcompleteness, the basis vectors are also non-orthogonal and the input-output relationship deviates from being purely linear. The justification of deviating from a strictly linear approach is to account for a weak form of nonlinearity exhibited by the simple cells themselves [94].

Dictionary learning: To design an overcomplete dictionary for the reference image $I_r \in \mathbb{R}^N$, a large number of distinct, possibly overlapping patches of dimension $\sqrt{n} \times \sqrt{n}$ are extracted *randomly* from I_r. Ideally, one patch centered at every pixel should be extracted; but in practice, extracting any large number of patches is sufficient for learning a good dictionary. After extracting a large number of random patches, the patches with low or no structural information are discarded (by removing the patches whose variance is zero or close to zero after mean removal). The remaining k patches are selected and each of the k image patches is converted to a vector of length n. These patch vectors are concatenated to form a matrix $\mathbf{B} \in \mathbb{R}^{n \times s}$.

Using the patches as input, we intend to learn a dictionary $\Phi = \{\phi_i\}_{i=1}^m$, $\phi_i \in \mathbb{R}^n$. We are interested in the *overcomplete* case where $m > n$ i.e. when Φ has more basis vectors than the dimensionality of the input. An overcomplete dictionary offers greater flexibility in representing the essential structures in a signal. It is also robust to additive noise, occlusion and small translation [1].

As discussed in Chapter 1, with overcompleteness however, greater difficulties arise; because a full-rank, overcomplete Φ creates an underdetermined system of linear equations having an infinite number of solutions. To narrow down the choice to one well-defined solution, constraints (e.g. minimum norm) are required. We enforce a constraint of sparsity in order to mimic the cortical model in [4]. Let the sparse representation of $\mathbf{B}$ over the dictionary Φ be denoted by $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^s$, $\mathbf{x}_i \in \mathbb{R}^m$ where any patch vector in $\mathbf{B}$ can be represented by a linear superposition of no more than k_1 dictionary columns where $k_1 << m$. This is formally written as

the following sparse optimization problem:

$$\min_{\{\Phi,\mathbf{X}\}} \quad \left\{ \|\mathbf{B} - \Phi\mathbf{X}\|_F^2 \right\} \quad \text{s. t. } \forall i \quad \|\mathbf{x}_i\|_0 \leq k_1 \tag{4.1}$$

where $\|.\|_F$ is the Frobenius norm (square root of the sum of the squared values of all elements in a matrix) and $\|.\|_0$ is the ℓ_0 semi-norm that counts the number of non-zero elements in a vector. To solve Equation 4.1, a popular learning algorithm, known as the K-SVD [23] is employed. K-SVD iteratively solves Equation 4.1 by performing two steps at each iteration: (i) *sparse coding* and (ii) *dictionary update*. In the sparse coding step, Φ is kept fixed and the coefficients in $\mathbf{X}$ are computed by the greedy algorithm OMP [12]:

$$\min_{\mathbf{X}} \quad \left\{ \|\mathbf{B} - \Phi\mathbf{X}\|_F^2 \right\} \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq k_1 \tag{4.2}$$

In the dictionary update step, each basis vector $\phi_i \in \Phi$ is updated sequentially, allowing the corresponding coefficients in $\mathbf{X}$ to change as well.

4.2.2 The quality estimation phase

To estimate the quality of an image, we compare the local patches of this image with the corresponding patches in its reference image. Instead of using all possible image patches (as in SSIM and its variants), we intend to compare only a set of carefully selected image patches. These selected patches are considered to be visually more important than others. Later (see Section 3.3 andFigure 4.4), we also show that higher quality scores are obtained using the visually important patches as opposed to the scores obtained using all image patches. A global measure of quality is then computed by aggregating the scores obtained at the local level.

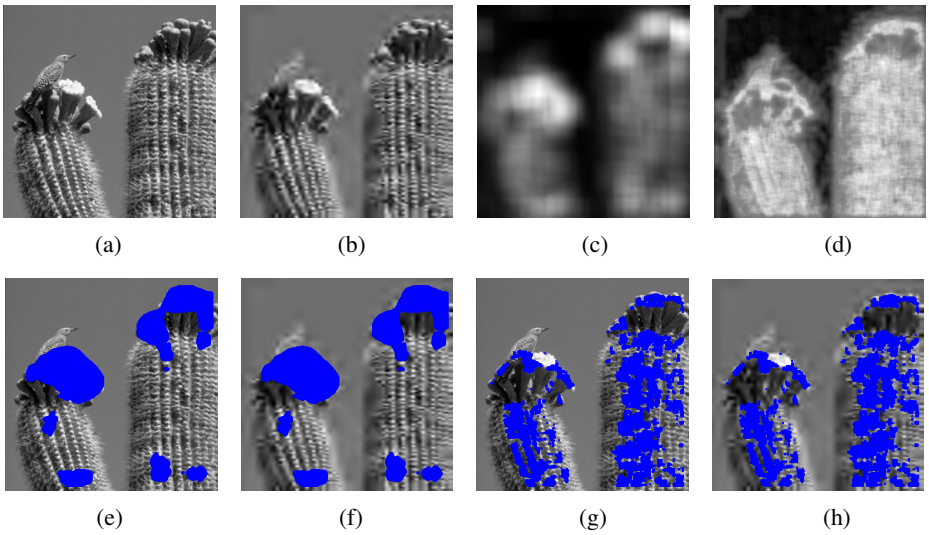


Figure 4.2: (a) Reference image, (b) distorted image, (c) combined saliency map using spectral residual method (d) combined local entropy map, (e) visually important pixels in the reference image detected based on spectral residual, (f) corresponding pixels in the distorted image detected based on spectral residual, (g) visually important pixels in the reference image detected based on entropy, (h) corresponding pixels in the distorted image based on entropy. (Note that the displayed images are smaller than the original and human perception of important regions may vary with image size. The image is best viewed in color.)

Detection of the visually important patches

It is well-known that not every pixel (or region) in an image receives the same level of visual importance. Several studies have shown that a significant improvement in the performance of quality metrics can be achieved by detecting the perceptually important regions [96–98].

In order to detect the visually important regions in an image, any visual saliency detection method can be used. In this work, we experiment with the following salient patch detection methods:

a) Itti-Koch saliency model [99]

b) Graph-based visual saliency [100]

c) Spectral residual [101]

d) Entropy-based method

The first three methods mentioned above are well known saliency detection methods and the last one is a rather simple approach. Our experiments show that the c) spectral residual and the d) entropy-based methods yield the best results for our purpose (details in section 4.3.3). A brief description of each of the four salient patch detection methods is provided below.

Itti-Koch saliency model: This classical method decomposes the input image into a set of feature maps by extracting multiple low level features (such as intensity, color and orientation) at different scales. These feature maps are normalized and combined across scales to form conspicuity maps, one for each feature. These conspicuity maps are then combined to create one saliency map of the image. For details, please refer to the work of Itti et al. [99].

The *Graph-based visual saliency* (GBVS): This successful method uses the computational power and the parallel nature of the graph algorithms to compute saliency map of images. Like [99], GBVS also computes multiple feature maps in order to find out the points that are unusual in its neighborhood using a graph-based algorithm. The maps that identify unusual points are called the activation maps. These activation maps are normalized and combined to create the saliency map. Details can be found in the original reference [100].

The *Spectral Residual approach*: This is a state-of-the-art saliency detection

method that can compute robust saliency map of natural images very fast. This method analyzes the frequency spectrum of an input image obtained by the Fourier transform. The method extracts the points of statistical singularities in the spectrum which corresponds to the salient regions in the spatial domain. For details, please refer to the original work [101].

Entropy-based visually important patch detection: A common hypothesis is that the HVS is an efficient extractor of information, and therefore the image regions that contain high information attract more visual attention [87, 85]. Based on this hypothesis, we take an information theoretic approach towards detecting the visually important patches. One way to quantify the local information content of an image is by computing the Shannon's entropy of each patch. The information content or entropy of a discrete random variable $\mathbf{z}$ with probability distribution $\mathbb{P}_z = \{\mathtt{p}_1, \mathtt{p}_2, ..., \mathtt{p}_J\}$ is defined as

$$H(\mathbf{z}) = H(\mathbb{P}_z) = -\sum_{j=1}^{J} \mathtt{p}_j \log_2 \mathtt{p}_j \tag{4.3}$$

Similarly, an image patch can also be analyzed as a random variable. Let us consider an image patch $\mathbf{z}$ of dimension $\sqrt{n} \times \sqrt{n}$ where each pixel in $\mathbf{z}$ is assumed to be independent and identically distributed. If $\mathbf{z}$ contains J distinct intensity values, its probability distribution, $\mathbb{P}_z$, is given by $\mathbb{P}_z = \{\mathtt{p}_1, \mathtt{p}_2, ..., \mathtt{p}_J\}$, where $J \leq 2^8$ for an 8-bit grayscale image; p_j is the probability of the pixel intensity value j. The probability $\mathtt{p}_j$ is defined as $\mathtt{p}_j = f_j/n$, where f_j is the number of pixels (frequency) with intensity value j occuring in the image patch $\mathbf{z}$ and n is the total number of pixels in $\mathbf{z}$. The entropy of every $\sqrt{n} \times \sqrt{n}$ patch (a patch around every pixel) in the reference image $I_r \in \mathbb{R}^N$ is computed as

$$H(\mathbf{z}) = -\sum_{j=1}^{J} \mathtt{p_j} \log_2 \mathtt{p_j} = -\frac{1}{n} \sum_{j=1}^{J} f_j \log_2 (f_j/n) \tag{4.4}$$

The larger the value of H, the higher is the information content of a patch.

Let us denote the saliency maps pertaining to I_r and I_d by M_r and M_d, computed by one of the four saliency detection methods mentioned above. A *combined saliency map* of the same size as M_r and M_d is then created as $max(M_r, M_d)$ i.e. by taking the maximum of the values of M_r and M_d at each point. The locations of the pixels in the combined map that have high saliency scores are found. These points are used to select the corresponding q patches from I_r and I_d as the *visually important patches* (see Figure 4.2 for details). These patches upon extraction from I_r and I_d are vectorized and arranged in columns of the matrices $\mathbf{B}_r \in \mathbb{R}^{n \times \mathtt{q}}$ and $\mathbf{B}_d \in \mathbb{R}^{n \times \mathtt{q}}$ respectively.

Computation of the quality score

At this point, we have two sets of visually important patches: $\mathbf{B}_r$ and $\mathbf{B}_d$, extracted from the same locations in the reference and the distorted images. The next step is to compare these patches w.r.t. the dictionary Φ.

Let us consider any patch vector $\mathbf{b}_r \in \mathbf{B}_r$ from I_r and its corresponding patch vector $\mathbf{b}_d \in \mathbf{B}_d$ from I_d. The patches $\mathbf{b}_r$ and $\mathbf{b}_d$ are decomposed using Φ to obtain their respective sparse coefficient vectors $\mathbf{x}_r$ and $\mathbf{x}_d$.

$$\min_{\mathbf{x}_r} \quad \left\{ \|\mathbf{b}_r - \Phi \mathbf{x}_r\|_2^2 \right\} \quad \text{subject to} \quad \|\mathbf{x}_r\|_0 \leq k_2 \tag{4.5}$$

$$\min_{\mathbf{x}_d} \quad \left\{ \|\mathbf{b}_d - \Phi \mathbf{x}_d\|_2^2 \right\} \quad \text{subject to} \quad \|\mathbf{x}_d\|_0 \leq k_2 \tag{4.6}$$

Note that, each of $\mathbf{x}_r$ and $\mathbf{x}_d$ contains only k_2 non-zero elements. The locations (indices) of these non-zero coefficients indicate those specific basis vectors in Φ which actually contribute to the approximation of the input patch. These active basis vectors are called the *support* of the input. The amplitudes of these non-zero coefficients are the weights by which these support vectors are combined. The support vectors and their weights together are indicative of the structural and non-structural distortions (e.g. luminance or contrast change) between the two input patches. Ideally, $\mathbf{b}_d$ and $\mathbf{b}_r$ would have different sets of support vectors whenever there exist any structural distortions between them. Otherwise, if the two patches undergo purely non-structural distortions, the supports would remain the same but their weights may change.

In order to quantify the perceptual quality of $\mathbf{b}_d$ w.r.t. $\mathbf{b}_r$, we compare their sparse representations $\mathbf{x}_d$ and $\mathbf{x}_r$. A simple but effective way to compare two vectors is to compute their *normalized correlation coefficient*. A parameter α is computed based on the correlation coefficient between $\mathbf{x}_r$ and $\mathbf{x}_d$ as follows:

$$\alpha(\mathbf{b}_r, \mathbf{b}_d) = \frac{\left|\mathbf{x}_r^T \mathbf{x}_d\right| + c_1}{\|\mathbf{x}_r\|_2 \|\mathbf{x}_d\|_2 + c_1} \tag{4.7}$$

where c_1 is a small positive constant added to avoid instability when the denominator is close to zero. Clearly, $0 < \alpha \leq 1$. When $\mathbf{x}_r$ and $\mathbf{x}_d$ are orthogonal, $\left|\mathbf{x}_r^T \mathbf{x}_d\right| = 0$; but due to the presence of c_1, the parameter α is slightly greater than zero. Due to normalization, α is unaffected by the lengths of $\mathbf{x}_r$ and $\mathbf{x}_d$. Thus α is unable to measure distortions that cause the length of $\mathbf{x}_d$ to change. To account for these types of distortions as well, we introduce another parameter. An important measure of similarity (or difference) between two vectors is their *pointwise difference*.

Hence, we compute another quantity β which uses the length of the difference vector $(\mathbf{x}_r - \mathbf{x}_d)$.

$$\beta(\mathbf{b}_r, \mathbf{b}_d) = 1 - \frac{\|\mathbf{x}_r - \mathbf{x}_d\|_2 + c_2}{\|\mathbf{x}_r\|_2 + \|\mathbf{x}_d\|_2 + c_2} \tag{4.8}$$

where c_2 is a small positive constant. It is easy to see that $0 < \beta < 1$, for non-empty $\mathbf{x}_r$ and $\mathbf{x}_d$.

We propose a function $\mathscr{S}(\mathbf{b}_r, \mathbf{b}_d)$ that measures the perceptual quality of $\mathbf{b}_d$ w.r.t $\mathbf{b}_r$ as follows:

$$\mathscr{S}(\mathbf{b}_r, \mathbf{b}_d) = \alpha(\mathbf{b}_r, \mathbf{b}_d)\beta(\mathbf{b}_r, \mathbf{b}_d) \tag{4.9}$$

Let $\mathscr{S}(\mathbf{b}_r^i, \mathbf{b}_d^i)$ be the quality measure of $\mathbf{b}_d^i$- the ith salient patch in I_d, w.r.t. $\mathbf{b}_r^i$ - the corresponding patch in I_r. The proposed *global* image quality $\text{SPARQ}(I_r, I_d)$ is computed by averaging over all $\mathtt{q}$ visually important patches.

$$\text{SPARQ}(I_r, I_d) = \frac{1}{\mathtt{q}} \sum_{i=1}^{\mathtt{q}} S(\mathbf{b}_r^i, \mathbf{b}_d^i) \tag{4.10}$$

Remarks:

- The SPARQ index (in Equation 4.10) is bounded: $0 < \text{SPARQ} < 1$; it is always non-negative since each of its components is non-negative.
- The highest value of SPARQ is attained when $I_r = I_d$.
- The index is *not* symmetric i.e. $\text{SPARQ}(I_r, I_d) \neq \text{SPARQ}(I_d, I_r)$. This is because the dictionary Φ is trained on the reference image only. Symmetry can be achieved by repeating the quality estimation stage with a dictionary trained on the distorted image and averaging the resulting quality scores obtained using the two dictionaries. Our experiments show that achieving sym-

metry has little or no significance on the performance of the SPARQ index.

4.3 Experimental validation

This section presents a critical evaluation of the proposed image quality metric, the SPARQ index, on the six publicly available image databases whose subjective quality ratings are available. The images in these databases contain a variety of distortions such as compression artifacts, blurring, flicker noise, wireless transmission artifacts, etc. First, we discuss how to set the parameter values required to compute SPARQ index. Experiments are carried out to select the salient patch detection method for which SPARQ performs the best. The performance of a quality metric is evaluated by computing the correlation between its objective scores and the available subjective ratings. To compare the performance of SPARQ with state-of-the-art, correlation scores of SPARQ are compared with those of the seven well-known image quality metrics: PSNR, SSIM [82], PSNR with HVS properties (PHVS-M) [102], Information Fidelity Criterion (IFC) [81], Visual Information Fidelity (VIF) [69], Visual Signal to Noise Ratio (VSNR) [79] and Information Weighted Structural Similarity (IWSSIM) [85].

4.3.1 The databases

A brief description of each of the six datasets used in this work is provided below.

The *Cornell-A57* dataset [79, 88] consists of 54 distorted images created from 3 original grayscale images. The images are subject to the following 6 types of distortions: JPEG compression, JP2K compression, AWGN, Gaussian blur, JPEG2000 compression with dynamic contrast-based quantization algorithm, and uniform quantization of LH subbands of a 5-level discrete wavelet transform at all

scales.

The *CSIQ* database [89] has 30 original images which were used to create 866 distorted images. The 6 distortion types (at four to five distortion levels) include JPEG compression, JP2K compression, global contrast decrements, AWGN, and Gaussian blurring.

The *LIVE* database [82, 90] contains 779 distorted images created from 29 original color images. Each distorted image exhibits one of the five types of distortions: JPEG2000 compression (JP2K), JPEG compression (JPEG), additive white gaussian noise (AWGN), Gaussian blur and fastfading channel distortion of JPEG2000 compressed bitstreams.

The *MICT-Toyoma* database [91] contains 168 distorted images created from 14 reference images. The images exhibit 2 types of distortions: JPEG and JP2K compression.

The *TID* database [92] is so far the largest subject-rated image dataset for quality evaluation. It has 1700 images generated from 25 reference images with 17 distortion types at four distortion levels. The distortion types are: AWGN, additive noise in color components, spatially correlated noise, masked noise, high frequency noise, impulse noise, quantization noise, Gaussian blur, image denoising, JPEG compression, JP2K compression, JPEG transmission errors, JP2K transmission errors, non-eccentricity pattern noise, local block-wise distortions of different intensity, mean shift, and contrast change.

The *WIQ* database [93, 103] consists of 80 distorted images generated from 7 reference images. The images exhibit wireless imaging artifacts which are not considered in other datasets. Due to the complex nature of a wireless communication channel, the images contain more than one artifacts.

4.3.2 Evaluation methodology

The results of an objective image quality assessment metric is compared with the subjective scores using a set of evaluation measures suggested by the VQEG [104]. These evaluation measures are - the Spearmans Rank Order Correlation Coefficient (SROCC), the Kendall's Rank Order Correlation Coefficient (KROCC), the Pearson Linear Correlation Coefficient (CC), Mean Absolute Error (MAE) and Root Mean Squared Error (RMS). The SROCC and KROCC are used to measure the *prediction monotonicity*, while CC, MAE and RMS measure the *prediction accuracy* of the objective scores. In order to compute CC, MAE and RMS, a five-parameter logistic function (refer to Equation 4.11 and Equation 4.12) is fitted to the objective scores. A particular objective score, S, is mapped to a new score, $\mathscr{Q}(S)$ using a non-linear mapping function $\mathscr{Q}(\cdot)$ which is defined as follows.

$$\mathscr{Q}(S) = \varphi_1 \text{logistic}(\varphi_2, (S - \varphi_3)) + S\varphi_4 + \varphi_5 \tag{4.11}$$

$$\text{logistic}(\sigma, S) = \frac{1}{2} - \frac{1}{1 + \exp(\sigma, S)} \tag{4.12}$$

A MATLAB function called `fminunc` is used for fitting. The values of CC, MAE and RMS are computed *after* performing the above non-linear mapping between the subjective and objective scores. Note that, SROCC and KROCC are non-parametric rank correlation metrics and are independent of any nonlinear mapping between the subjective and the objective scores. A good image quality assessment metric is expected to have high SROCC, KROCC and CC scores, and low MAE and RMS values. For details of the evaluation methodology please see the original works [69, 85, 104].

4.3.3 Implementation details

Preprocessing

Before training and quality assessment, two preprocessing steps are executed: (1) every color image in each dataset is converted to grayscale image, and (2) all images (reference and distorted) is downsampled by a factor $\mathscr{F}$ so as to account for the viewing condition. The value of F is obtained by using the following empirical formula [82].

$$\mathscr{F} = \max(1, \text{round}(\text{g}/256)) \tag{4.13}$$

where $\text{g} = \min(\#\text{rows in } I_{ref}, \#\text{columns in } I_{ref})$.

Training

In the training phase, there are 4 parameters to be set:

- $\sqrt{n}$: patch size
- s : number of patches to be extracted from a reference image for training the dictionary
- m : number of basis vectors in the dictionary
- k_1 : sparsity constraint

Unfortunately, there is no theoretical guidelines to determine the values of these parameter, so we rely on previous work and empirical methods. A patch size of $\sqrt{n} \times \sqrt{n} = 11 \times 11$ is used following the patch-size specification of SSIM [82]. A collection of as large as $s = 3000$ patches are extracted *randomly* from every reference image to train its corresponding dictionary. We set the overcompleteness

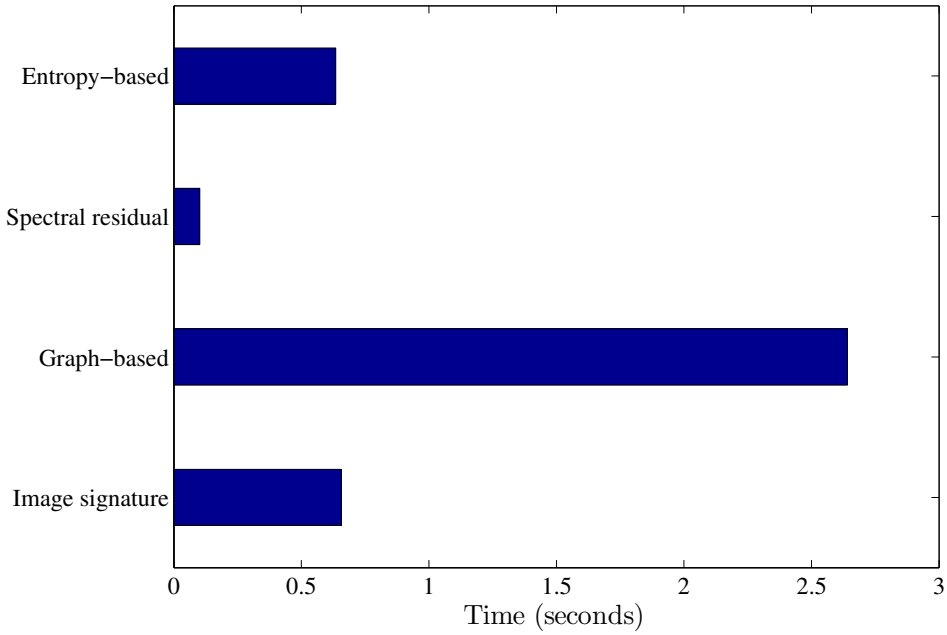


Figure 4.3: Comparison of the 4 visually important patch detection methods in terms of computation time for the same pair of images (size 256×256).

factor (m/n) to 2 which yields $m = 242$. It has been shown that for low overcompleteness factor, sparse representations are stable in the presence of noise [68]. The value of k_1 is set to 12 which is approximately 10% of the dimensionality of the input vectors.

Selecting the visually important regions

As mentioned before, 3 popular saliency methods (image signature [99], graph-based visual saliency [100], spectral residual [101]) and a simple entropy-based approach are considered to detect the visually important patches from images. In order to investigate the effect of these methods on quality assessment, we employed each method within our framework and observe their performance on the

Table 4.1: Performance comparison of the visually important patch detection methods when used to compute SPARQ. The performances are evaluated in terms of SROCC scores.

Patch selection method	A57	CSIQ	LIVE	MICT	TID	WIQ
Random	0.875	0.904	0.870	0.766	0.674	0.778
Itti-Koch saliency [99]	0.926	0.941	0.915	0.848	0.805	0.800
Graph-based [100]	0.909	0.939	0.914	0.865	**0.806**	0.807
Spectral residual [101]	0.920	0.946	0.930	**0.872**	0.792	0.816
Entropy-based	**0.943**	**0.950**	**0.933**	0.870	0.774	**0.851**

six datasets.

Figure 4.3 compares the 4 patch selection methods in terms of computation time. Table 4.1 compares them in terms of SROCC indicating their impact on the quality assessment method. Each method uses the same number of q salient patches (see Section 4.3.3 for how to determine the value of q). The result of selecting *random* patches is also presented in Table 4.1. Random patch selection result serves as a baseline.

It is clear from Table 4.1 that carefully selecting and using the visually important patches for quality assessment improve the performance of the proposed quality metric. Another important observation is that the simple entropy-based method performs better or at par with the well-known saliency methods in the context of quality assessment.

Considering the performance (refer to Table 4.1) and speed (refer to Figure 4.3) of the competing patch detection methods, we observe that the spectral residual method and the entropy-based approach are the two better methods. From this point, we will use the following notations of SPARQ depending on which patch selection method it uses:

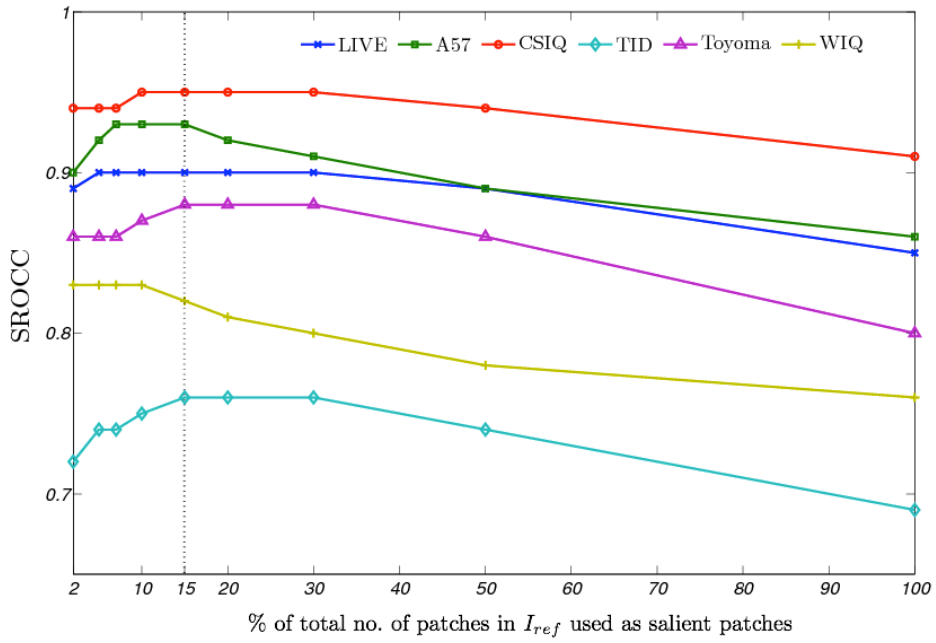


Figure 4.4: Performance of the SPARQ$_e$ index (correlation with subjective scores measured in terms of SROCC) varies with the percentage of high-entropy patches used in the quality estimation process.

- SPARQ$_e$ - uses entropy for patch selection
- SPARQ$_{sr}$ - uses spectral residual for patch selection.

Note that, other than the patch detection method all parameters remain the same.

Quality estimation

In the quality estimation phase, we need to set the following parameters:

- c_1, c_2 : stabilizing constants in Equation 4.7 and Equation 4.8
- q : number of salient patches

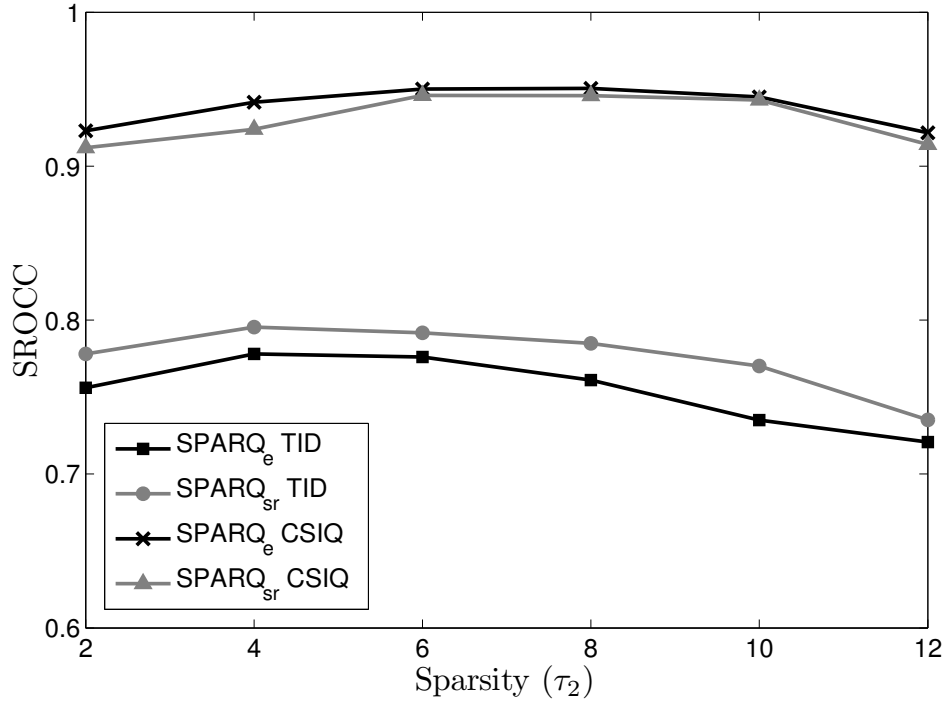


Figure 4.5: Effect of sparsity on the performances of SPARQ$_e$ and SPARQ$_{sr}$ on TID and CSIQ datasets

- k_2 : sparsity constraint

The constants are chosen to have very small positive values, $c_1 = 256 * 0.01$, $c_2 = 0.01$ so as to have minimal influence on the quality score. The value of q is determined empirically. For each database, the number of salient patches, q, is varied and the performance of SPARQ is measured in terms of the correlation between its scores and the subjective scores. This is presented in Figure 4.4 where the Spearmans Rank Order Correlation Coefficient (SROCC) is plotted against q. The value of q is varied from 2% to 100% of N where N is the total number of patches (one around each pixel) in I_r or I_d. In five out of the six datasets, the best performance of the SPARQ index is observed when $\mathrm{q} = 0.15N$ i.e. 15% of N. Also

Table 4.2: Performance comparison of various quality assessment metrics over six datasets

SROCC-based comparison									
Dataset	PSNR	SSIM	PHVS-M	IFC	VIF	VSNR	IWSSIM	SPARQ$_e$	SPARQ$_{sr}$
A57	0.598	0.806	0.896	0.318	0.622	**0.935**	0.775	**0.943**	0.919
CSIQ	0.800	0.858	0.822	0.767	0.919	0.809	0.921	**0.950**	**0.946**
LIVE	0.875	0.947	0.922	0.926	**0.963**	0.912	**0.956**	0.933	0.930
MICT	0.613	0.875	0.848	0.835	**0.907**	0.860	**0.920**	0.870	0.871
TID	0.552	0.773	0.561	0.622	0.749	0.704	**0.853**	0.774	**0.792**
WIQ	0.626	0.758	0.757	0.716	0.692	0.656	0.786	**0.851**	**0.816**
PLCC-based comparison									
Dataset	PSNR	SSIM	PHVS-M	IFC	VIF	VSNR	IWSSIM	SPARQ$_e$	SPARQ$_{sr}$
A57	0.628	0.802	0.875	0.372	0.614	0.914	0.765	**0.945**	**0.925**
CSIQ	0.746	0.758	0.772	0.821	0.927	0.735	0.914	**0.945**	**0.939**
LIVE	0.860	0.941	0.917	0.853	**0.944**	0.917	**0.951**	0.930	0.928
MICT	0.632	0.705	0.839	0.833	**0.902**	0.855	0.802	**0.873**	0.872
TID	0.519	0.727	0.552	0.660	0.808	0.682	**0.851**	0.805	**0.820**
WIQ	0.639	0.640	0.749	0.705	0.730	0.763	0.660	**0.836**	**0.801**
RMS-based comparison									
Dataset	PSNR	SSIM	PHVS-M	IFC	VIF	VSNR	IWSSIM	SPARQ$_e$	SPARQ$_{sr}$
A57	0.191	0.147	0.119	0.223	0.194	0.099	0.105	**0.080**	**0.093**
CSIQ	0.175	0.171	0.167	0.150	0.098	0.178	0.150	**0.086**	**0.090**
LIVE	13.990	9.985	10.892	14.263	**9.240**	10.772	**8.347**	10.016	10.185
MICT	0.969	0.887	0.680	0.692	**0.540**	0.648	0.748	0.611	0.612
TID	1.147	0.921	1.119	1.008	0.790	0.981	**0.689**	0.796	**0.768**
WIQ	15.426	17.595	15.185	16.252	15.653	14.809	17.208	**12.552**	**13.699**

notice that, when all patches in I_r are used, the performance of the SPARQ index degrades. This confirms our assumption that only the visually important areas are useful for quality assessment. For all datasets, we use the same parameter values.

In order to determine the value of k_2, it is varied from 2 to 12, and the changes in SROCC scores are plotted in Figure 4.5 for SPARQ$_e$ and SPARQ$_{sr}$. The results are shown for the larger datasets available: the TID and CSIQ datasets. From Figure 4.5, we see that $k_2 = 6$ provides the best overall trade-off.

4.3.4 Performance comparison

Table 4.2 compares the performance of SPARQ$_e$ and SPARQ$_{sr}$ with the state-of-the-art quality metrics in terms of SROCC, CC and RMS (KROCC and MAE are left out for simplicity and because they reflect similar performance trends as SROCC and

Table 4.3: Overall performance comparison of image quality assessment algorithms

	SROCC		CC	
Quality metric	Direct avg.	Weighted avg	Direct avg.	Weighted avg
PSNR	0.677	0.685	0.670	0.655
SSIM [82]	0.836	0.835	0.762	0.778
PHVS-M [102]	0.801	0.722	0.784	0.704
IFC [81]	0.697	0.729	0.707	0.744
VIF [69]	0.809	0.839	0.821	0.865
VSNR [79]	0.813	0.783	0.811	0.758
IWSSIM [85]	0.868	**0.891**	0.824	**0.879**
SPARQ$_e$	**0.887**	0.858	**0.889**	0.871
SPARQ$_{sr}$	**0.879**	**0.864**	**0.881**	**0.875**

RMS, respectively). PSNR is used as a baseline method. For the implementation of SSIM, PHVS-M, IFC, VIF, VSNR and IWSSIM we have used the original MATLAB codes provided by the respective authors. The parameters of each of these methods are set to their default values as suggested in the original references.

The best two results in Table 4.2 are written in bold for each dataset. As can be seen in the comparison, no single metric performs the best on all datasets. Nevertheless, the performances of SPARQ$_e$ and SPARQ$_{sr}$ are consistently high over all datasets.

In order to provide a bigger picture, the average SROCC and CC values are computed over all six datasets in Table 4.3. The average values are computed for two cases: in the first case the values are *directly* averaged and in the second case the values are *weighted* by the size of the databases. The weight for a particular database is the number of distorted images it contains, e.g. 779 for LIVE and 54 for A57. In each case, the best two results are printed in boldface. The performance of

the SPARQ index for separate distortion types is presented in Table 4.4.

Table 4.4: Performance of SPARQ Index for different distortion types

JPEG					
Database	SROCC	KROCC	CC	MAE	RMS
A57	0.983	0.944	0.971	0.054	0.061
CSIQ	0.971	0.850	0.986	0.039	0.051
LIVE	0.970	0.851	0.978	5.109	6.684
MICT	0.859	0.668	0.864	0.512	0.622
TID	0.919	0.730	0.943	0.399	0.565
JPEG 2000					
Database	SROCC	KROCC	CC	MAE	RMS
A57	0.983	0.944	0.955	0.060	0.066
CSIQ	0.979	0.883	0.985	0.041	0.054
LIVE	0.943	0.790	0.951	5.849	7.790
MICT	0.928	0.770	0.927	0.378	0.462
TID	0.966	0.840	0.973	0.360	0.447
AWGN					
Database	SROCC	KROCC	CC	MAE	RMS
A57	0.967	0.889	0.973	0.023	0.030
CSIQ	0.962	0.836	0.961	0.033	0.046
LIVE	0.975	0.866	0.980	4.380	5.515
TID	0.756	0.546	0.742	0.309	0.409
Gaussian Blur					
Database	SROCC	KROCC	CC	MAE	RMS
A57	0.916	0.778	0.953	0.045	0.060
CSIQ	0.978	0.873	0.981	0.042	0.055
LIVE	0.947	0.799	0.941	4.730	6.249

Continued on next page

Table 4.4 – continued from previous page

Database	SROCC	KROCC	CC	MAE	RMS
TID	0.947	0.803	0.941	0.297	0.397

Remarks:

- SPARQ clearly outperforms well-known quality metrics like SSIM, VSNR and VIF.
- SPARQ achieves highest correlation score in 3 out of the 6 datasets.
- SPARQ is among the top two performers in 4 out of the 6 datasets but does not perform very well on the LIVE dataset.
- Table 4.3 shows that on average, SPARQ_{sr} is slightly better than SPARQ_{e}.
- Overall, SPARQ is always among the top two performing metrics with IWSSIM being its closest rival (see Table 4.3). However, it is important to note that IWSSIM is a multi scale method, while SPARQ operates on a single scale.
- The WIQ dataset is the only dataset that contains more than one artifacts due to the nature of wireless imaging. Notice that, SPARQ handles such complex artifacts much better than any other metric. This indicates the potential of SPARQ index to be used in complex practical systems where degradation of images is likely to be caused by more than one factors.
- The high correlation scores of SPARQ_{e} presented in Table 4.4 show that SPARQ is capable of handling different distortions.

4.3.5 Computational complexity

In order to compute the SPARQ index, the two steps that require the bulk of computation are (i) the dictionary learning step in the training phase and (ii) the sparse coding step in the quality estimation phase. The computational load of the dictionary learning step in turn is dominated by the sparse coding step performed as part of the learning process. Hence, it is the sparse coding step that we should be concerned with.

Our implementation uses an efficient sparse coding algorithm called the *Batch-OMP* [52]. Its computational complexity is $\mathcal{O}(mnk)$ per training signal, where the dictionary dimension is $m \times n$ and k is the sparsity constraint and $s << n$ [52].

To give an idea of the computation time, a basic MATLAB implementation (on a computer with Intel Q9400 processor at 2.66 GHz) takes on average 3.4 seconds for the dictionary learning step using the parameter values specified in this paper. The quality estimation step for SPARQ_e takes 1.7 sec and for SPARQ_{sr} it take 1.0 sec.

4.3.6 Limitations of SPARQ

Due to its dependence on sparse coding (Equation 4.2, Equation 4.5, Equation 4.6), SPARQ is computationally demanding (still much less expensive compared to the HVS-based models like MAD [105]). Nevertheless, considering the rapid growth of the area, we are hopeful that faster sparse coding algorithms will be available soon.

The present version of SPARQ index works on grayscale images and thus is blind to the degradations in the color components. Like most of the existing image quality assessment metrics, SPARQ relies on fidelity to quantify perceptual quality

where fidelity is one of the several factors in determining the perceptual quality [106].

4.4 Summary

In this chapter, we develop a new metric, the SPARQ index, that estimates the perceptual quality of a distorted image with respect to a reference image. This metric measures the structural delity between a reference image and its distorted versions. The performance of the SPARQ index is shown to be consistently better or comparable to the state-of-the-art quality metrics such as IWSSIM and VIF. The success of SPARQ is attributed to a new framework proposed in this chapter. The framework is designed to extract the perceptually meaningful structural information from images, by learning overcomplete dictionaries.

Chapter 5

Conclusions

In this thesis, we have explored the usefulness of sparse representations obtained by learning overcomplete dictionaries for (i) image and video classification, (ii) image similarity measurement, and (ii) perceptual image quality assessment. Each of these problems is critically important to modern information processing systems and requires compatibility with human visual perception. We have addressed each problem separately from a perspective that aims to improve on their respective state-of-the-art. We have been able to achieve encouraging results in every case. This chapter clearly and concisely lists the contributions of our work and the possible directions of future work.

5.1 Contributions

This section summarizes the contributions of this thesis, indicated separately for each of three problems studied.

Sparse representation-based classification

- This is one of the pioneering works that explore the usefulness of learning sparse representations for classification. To the best of our knowledge, this is also the first work to propose a sparse representation-based approach to address the problem of human action recognition in videos.

- We have studied three dictionary learning frameworks: shared, class-specific and concatenated.

  - The usage of shared dictionaries learnt using vector quantization is common. We have shown that shared dictionaries learnt using sparse representations are more effective in classification. Our experiments show that sparse representation-based approach improves the recognition accuracy by 7% on the UCF sports dataset.

  - We have successfully employed the less-known class-specific framework and shown that this framework yields superior classification accuracy compared to the well-known shared dictionary framework on all datasets.

  - The concatenated framework is introduced in this thesis. This framework also has been shown to produce better results compared to those of the shared framework on all datasets.

- The proposed classification algorithms (Shared-hist, RSR, EFVC and Concat) have been shown to consistently perform better or at par with the state-of-the-art. Their robustness against partial occlusion, spatio-temporal scale variations, moderate viewpoint changes also has been proved.

- The proposed classification algorithms are fairly general and are applicable to a wide variety of image and video-based classification problems. Chapter 2 demonstrates successful applications of these algorithms for image-based face recognition and video-based human action recognition. Additional results on video-based facial expression recognition and image-based biological species classification are provided in the Appendix B. Our classification algorithms should also be applicable to other datatypes such as audio signals.

- We have been shown in our experiments that sparse representation outperforms (more than 10% higher recognition accuracy on the UCF sports dataset, refer to Table 2.6) the well-established vector quantization-based approach to learning dictionaries for classification.

- We have also proposed a simple and effective method for detecting and computing important motion patterns from videos. This method is called the Local Motion Pattern (LMP) descriptor. LMP is 3-4 times faster (refer to Table C.1) than state-of-the-art methods like Cuboids, and in general, is suitable for systems where speed is more important than accuracy. Nevertheless, for simple video datasets we have shown that LMP can outperform Cuboids.

Sparse representation-based image similarity measurement

- A sparse representation-based approach for computing an image similarity measure is introduced. The proposed measure is generic, in the sense that it assumes no prior knowledge of the data or the application.

- For the first time, we have identified an important connection between sparse

representation and the theoretical measure of data complexity, namely the Kolmogorov complexity. We exploit this connection to extend the areas of Kolmogorov complexity-based similarity measurement. We hope that the connection identified in our work will stimulate interest in the area of similarity measurement using Kolmogorov complexity and sparse representation.

- The previously developed Kolmogorov complexity-inspired similarity methods were not very successful in the context of measuring image similarity. We have identified the issues with the previous approaches and proposed a sparse representation-based approach to compute image similarity. The proposed similarity measure has been shown to be successful in classifying, clustering and retrieving a variety of image data, such as textures, faces, biological species, etc.

Sparse representation-based perceptual image quality assessment

- We propose a new image quality metric - the Sparse Representation-based Quality (SPARQ) index. This quality metric measures the structural fidelity between the reference and the distorted image in order to quantify the visual quality of the distorted image.

- The SPARQ index is shown to consistently perform better or yield comparable results to the state-of-the-art. The success of SPARQ can be attributed to the proposed framework that extracts structural information in an image by using a model that mimics the response of the primary visual cortex to the stimuli.

- The proposed quality metric SPARQ relies on a model that mimics the re-

sponse of the primary visual cortex to visual stimuli. Hence, we expect that the structural information our method extracts from images is *perceptually meaningful*. This is an important advancement over the previous structural fidelity-based quality assessment methods which rely on only certain hypothesis regarding the operation of the HVS.

5.2 Future work

This thesis uses the K-SVD algorithm to learn dictionaries in all cases. This is because K-SVD is fast, simple to implement, and the most popular. Nevertheless, several improved dictionary learning methods have been proposed since the development of K-SVD in the last couple of years. Although these algorithms are not as simple as K-SVD, they can be employed to improve the accuracies at the cost of additional computational load. Similarly, more sophisticated solvers e.g. BP, FOCUSS can be employed to achieve better results but at the cost of higher computation time.

There are also several issue with K-SVD that we would like to point out as the possible directions of future work. Currently, no guideline is available on how to optimize the parameters in K-SVD. Systematic studies are required to understand the effect of each parameter on the dictionary learning process.

In general, using multiscale dictionaries can be a straightforward extension of all of the proposed methods.

Below, we discuss the possible directions to future work specific to each chapter.

Sparse representation-based classification

In this work, we have used Cuboids and the newly developed LMP features to obtain a rich representation of the action sequences. Features such as dense sampling, HoG3D, STIP [107], etc. can also improve the recognition accuracy, but usually are more expensive computationally. Similarly for image-based classification, features like SIFT is expected to improve the recognition accuracy.

The proposed classification algorithms disregard the spatial and temporal orientation of the extracted features. Incorporating the information regarding the location of the features will result in improvement of the current results. This can even help in detecting and recognizing the multiple actions in a single video.

Hierarchical dictionaries and discriminative dictionaries [24] can be useful for classification. Also building hybrid dictionaries that use a combination of multiple features is also worth studying.

Sparse representation-based image similarity measurement

Our work did not study speeding up the classification, retrieval or the clustering processes. This is because our objective was to demonstrate the usefulness and generality of the new distance measure. Further research can be done on how to use the proposed measure more efficiently, especially when we need to classify or cluster larger datasets. This will require the employment of sophisticated machine learning techniques.

Applications can also be extended to problems such as multimedia copy detection and data mining.

Sparse representation-based perceptual image quality assessment

The quality metric developed in this chapter, can be easily applied to other problems involving similarity measurement such as image clustering. Because of its generic data-dependent approach, SPARQ is also suitable (may require minor modifications) for video signals.

The SPARQ index can be improved by combining it with various pooling strategies and by learning multiscale dictionaries. Another possible future work includes extending SPARQ to work for color images and videos.

Bibliography

[1] M.S. Lewicki and T.J. Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–365, 2000. → pages 2, 10, 71

[2] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Processing*, 15(12):3736–3745, Dec. 2006. → pages 2, 10, 11, 22, 68

[3] A.M. Bruckstein, D.L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009. → pages 2, 3, 8, 12

[4] 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. → pages 2, 9, 67, 70, 71

[5] Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37 (23):3311 – 3325, 1997. → pages 2, 9

[6] B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995. → pages 6

[7] JA Tropp, AC Gilbert, S. Muthukrishnan, and MJ Strauss. Improved sparse approximation over quasiincoherent dictionaries. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–37. IEEE, 2003. → pages 6

[8] D.L. Donoho, M. Elad, and V.N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *Information Theory, IEEE Transactions on*, 52(1):6–18, 2006. → pages 6

[9] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Processing*, 41(12):3397–3415, 1993. → pages 6

[10] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive approximation*, 13(1):57–98, 1997. → pages

[11] G.M. Davis, S.G. Mallat, and Z. Zhang. Adaptive time-frequency decompositions. *Optical Engineering*, 33(7):2183–2191, 1994. → pages

[12] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proc. Asilomar Signals, Systems and Computers*, 1993. → pages 6, 72

[13] S. Chen, S.A. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50(5):1873–1896, 1989. → pages 6

[14] V.N. Temlyakov. The best m-term approximation and greedy algorithms. *Advances in Computational Mathematics*, 8(3):249–265, 1998. → pages 6

[15] J.A. Tropp. Greed is good: Algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231–2242, 2004. → pages 7

[16] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20:33–61, 1998. ISSN 1064-8275. → pages 7

[17] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998. → pages 8

[18] I.F. Gorodnitsky and B.D. Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *Signal Processing, IEEE Transactions on*, 45(3):600–616, 1997. → pages 8

[19] S. Mallat. *A wavelet tour of signal processing: The sparse way, 3rd Ed.* Academic Press, NY, 2008. → pages 10

[20] M. Girolami. A variational method for learning sparse and overcomplete representations. *Neural computation*, 13(11):2517–2532, 2001. → pages 10

[21] K. Engan, S. O. Aase, and J. H. Husoy. Frame based signal compression using method of optimal directions (mod). In *Proc. ISCAS*, 1999. → pages 10, 11, 68

[22] M. Aharon, M. Elad, and A. Bruckstein. K-svd: Design of dictionaries for sparse representation. In *Proc. SPARS*, 2005. → pages

[23] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Processing*, 54:4311–4322, 2006. → pages 11, 21, 22, 50, 68, 72

[24] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proc. CVPR*, pages 1–8, 2008. → pages 11, 17, 25, 98

[25] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proc. ICML*, pages 689–696, 2009. → pages 10

[26] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Trans. Image Processing*, 17(1):53 –69, jan 2008. ISSN 1057-7149. → pages 10, 68

[27] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. PAMI*, 31:210–227, 2008. → pages 11, 12, 17, 31, 32, 116

[28] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007. → pages 12

[29] URL http://dsp.rice.edu/cs. → pages 12

[30] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. CVPR*, 2009. → pages 12, 17

[31] G. Peyré. Sparse modeling of textures. *J. Math. Imaging Vis.*, 34(1):17–31, 2009. → pages 12, 17, 25

[32] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. → pages 17, 18, 118

[33] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. ICCV VSPETS Workshop*, pages 65–72, 2005. → pages 19, 114, 115, 117, 121

[34] T. Guha and R.K. Ward. Action recognition by learnt class-specific overcomplete dictionaries. In *Proc. IEEE FG*, pages 143 –148, 2011. → pages 19

[35] Richard Baraniuk and Michael Wakin. Random projections of smooth manifolds. *Foundations of Computational Mathematics*, 9:51–77, 2009. → pages 20

[36] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981. → pages 25

[37] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proc. NIPS*, 2006. → pages 29

[38] Y.C. Eldar and H. Bolcskei. Block-sparsity: Coherence and efficient recovery. In *Proc. ICASSP*, pages 2885 –2888, april 2009. → pages 29

[39] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Trans. PAMI*, 29(12):2247–2253, Dec 2007. → pages 32, 35, 36, 37, 119

[40] M.D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Proc. CVPR*, pages 1–8, june 2008. → pages 32, 37, 39

[41] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proc. ACM Multimedia*, pages 357–360, 2007. → pages 35, 121

[42] Juan Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79:299–318, 2008. → pages 35

[43] Ziming Zhang, Yiqun Hu, Syin Chan, and Liang-Tien Chia. Motion context: A new representation for human action recognition. In *Proc. ECCV*, volume 5305, pages 817–829, 2008. → pages 35

[44] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *Proc. CVPR*, pages 1 –8, 2008. → pages 35

[45] Imran N. Junejo, Emilie Dexter, Ivan Laptev, and Patrick Perez. View-independent action recognition from temporal self-similarities. *IEEE Trans. PAMI*, 99, 2010. ISSN 0162-8828. → pages 35

[46] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Trans. PAMI*, 32(2):288 –303, Feb 2010. → pages 35

[47] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proc. BMVC*, Sep 2009. → pages 39, 40

[48] Yan Zhu, Xu Zhao, Yun Fu, and Yuncai Liu. Sparse coding on local spatial-temporal volumes for human action recognition. In *Proc. ACCV*, volume 6493, pages 660–671, 2010. → pages 39

[49] A. Yao, J. Gall, and L. Van Gool. A hough transform-based voting framework for action recognition. In *Proc. CVPR*, pages 2061 –2068, Jun 2010. → pages 39

[50] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Proc. CVPR*, pages 2046 –2053, june 2010. → pages 39

[51] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *Proc. ICCV*, pages 492 –497, Oct 2009. → pages 39

[52] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 2008. → pages 40, 91

[53] B. Girod. What's wrong with mean-squared-error? *Digital Images and Human Vision*, 1993. → pages 44, 65

[54] Ming Li, Xin Chen, Xin Li, Bin Ma, and P.M.B. Vitanyi. The similarity metric. *IEEE Trans. Information Theory*, 50(12):3250 – 3264, Dec 2004. → pages 44, 45, 47

[55] R. Cilibrasi and P.M.B. Vitanyi. Clustering by compression. *IEEE Trans. Information Theory*, 51(4):1523 – 1545, Apr 2005. → pages 44, 45, 46, 47

[56] T. Watanabe, K. Sugawara, and H. Sugihara. A new pattern representation scheme using data compression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):579 –590, may 2002. → pages 46

[57] A.J. Pinho and P.J.S.G. Ferreira. Image similarity using the normalized compression distance based on finite context models. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 1993 –1996, sept. 2011. → pages 47, 48

[58] B. J. L. Campana and E. J. Keogh. A compression-based distance measure for texture. *Statistical Analysis and Data Mining*, 3(6), 2010. → pages 44, 48, 53, 57, 58, 60, 61, 62, 116

[59] AN Kolmogorov. Three approaches to the quantitative definition ofinformation'. *Problems of information transmission*, 1(1):1–7, 1965. → pages 44

[60] Ray J Solomonoff. A formal theory of inductive inference. part i. *Information and control*, 7(1):1–22, 1964. → pages

[61] Gregory J Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM (JACM)*, 13(4):547–569, 1966. → pages 44

[62] Ming Li and Paul M.B. Vitnyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 2 edition, 1997. → pages 45

[63] Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards parameter-free data mining. In *Proc. ACM SIGKDD*, pages 206–215, 2004. → pages 46

[64] Rudi Cilibrasi, Paul M. B. Vitányi, and Ronald de Wolf. Algorithmic clustering of music based on string compression. *Computer Music Journal*, 28(4):49–67, 2003. → pages 46

[65] Xin Chen, B. Francia, Ming Li, B. McKinnon, and A. Seker. Shared information and program plagiarism detection. *IEEE Trans. Information Theory*, 50(7):1545 – 1551, july 2004. → pages 46

[66] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. → pages 53

[67] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. of Computer Vision*, 40: 99–121, 2000. ISSN 0920-5691. → pages 53

[68] B. Wohlberg. Noise sensitivity of sparse signal representations: reconstruction error bounds for the inverse problem. *IEEE Trans. Signal Processing*, 51(12):3053 – 3060, December 2003. → pages 55, 83

[69] H.R. Sheikh and A.C. Bovik. Image information and visual quality. *IEEE Tran. Image Processing*, 15(2):430 –444, feb. 2006. → pages 56, 66, 79, 81, 88

[70] URL http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. → pages 57

[71] URL http://cvc.yale.edu/projects/yalefaces/yalefaces.html. → pages 57

[72] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. NIPS*, pages 849–856. MIT Press, 2001. → pages 58

[73] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1998. → pages 58

[74] URL http://www.ux.uis.no/~tranden/brodatz.html. → pages 59

[75] URL http://www-cvr.ai.uiuc.edu/ponce_grp/data/. → pages 60

[76] URL http://www.nada.kth.se/cvap/databases/kth-tips/download.html. → pages 60

[77] Z. Wang and A. C. Bovik. *Modern Image Quality Assessment*. Morgan Claypool, 2006. → pages 65, 67

[78] Alesandr Shnayderman, Alexander Gusev, and Ahmet M. Eskicioglu. An svd-based gray-scale image quality measure for local and global assessment. *IEEE Tran. Image Processing*, 2006. → pages 66

[79] D.M. Chandler and S.S. Hemami. Vsnr: A wavelet-based visual signal-to-noise ratio for natural images. *IEEE Tran. Image Processing*, 16 (9):2284 –2298, sep 2007. → pages 66, 79, 88

[80] W. Lin and C.C.J. Kuo. Perceptual visual quality metrics: A survey. *J Visual Comm Image Representation*, 22(4):297 – 312, 2011. → pages 66

[81] Hamid Rahim Sheikh, Alan Conrad Bovik, and Gustavo de Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Tran. Image Processing*, 14(12):2117–2128, 2005. → pages 66, 79, 88

[82] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600 –612, Apr 2004. → pages 66, 79, 80, 82, 88

[83] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1398 – 1402, nov. 2003. → pages 67

[84] Guan-Hao Chen, Chun-Ling Yang, and Sheng-Li Xie. Gradient-based structural similarity for image quality assessment. In *ICIP*, pages 2929 –2932, oct. 2006. → pages 67

[85] Zhou Wang and Qiang Li. Information content weighting for perceptual image quality assessment. *IEEE Trans Image Processing*, 20(5):1185 –1198, may 2011. → pages 67, 75, 79, 81, 88

[86] Zhou Wang and E.P. Simoncelli. Translation insensitive image similarity in complex wavelet domain. In *ICASSP*, volume 2, pages 573 – 576, 18-23, 2005. → pages 66, 67

[87] Zhou Wang and Xinli Shang. Spatial pooling strategies for perceptual image quality assessment. In *ICIP 2006*, pages 2945 –2948, oct. 2006. → pages 67, 75

[88] URL http://foulard.ece.cornell.edu/dmc27/vsnr/vsnr.html. → pages 68, 79

[89] E.C. Larson and D. M. Chandler. Categorical image quality assessment (csiq) database. URL http://vision.okstate.edu/?loc=csiq. → pages 68, 80

[90] L. Cormack H.R. Sheikh, Z.Wang and A.C. Bovik. Live image quality assessment database release 2. URL http://live.ece.utexas.edu/research/quality. → pages 68, 80

[91] Y. Horita, K. Shibata, Y. Kawayoke, and Z.M.P. Sazzad. Mict image quality evaluation database. URL http://mict.eng.u-toyama.ac.jp/mictdb.html. → pages 68, 80

[92] N. Ponomarenko and K. Egiazarian. Tampere image database 2008 tid2008. URL http://www.ponomarenko.info/tid2008.htm. → pages 68, 80

[93] U. Engelke, T.M. Kusuma, H.J. Zepernick, and M. Caldera. Reduced-reference metric design for objective perceptual quality assessment in wireless imaging. *Signal Processing: Image Communication*, 24(7):525 –547, 2009. → pages 68, 80

[94] B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by vi? *Vision research*, 37(23):3311–3326, 1997. → pages 70

[95] David J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559 – 601, 1994. → pages 70

[96] E.C. Larson and D.M. Chandler. Unveiling relationships between regions of interest and image fidelity metrics. In *Visual Communications and Image Processing*, volume 6822, pages 68222A–68222A, 2008. → pages 73

[97] EC Larson, C. Vu, and DM Chandler. Can visual fixation patterns improve image fidelity assessment? In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2572–2575. IEEE, 2008. → pages

[98] U. Engelke, V.X. Nguyen, and H.J. Zepernick. Regional attention to structural degradations for perceptual image quality metric design. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 869–872. IEEE, 2008. → pages 73

[99] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. PAMI*, 20(11): 1254–1259, 1998. → pages 74, 83, 84

[100] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. 2007. → pages 74, 83, 84

[101] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. → pages 74, 75, 83, 84

[102] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin. On between-coefficient contrast masking of dct basis functions. In *Int. Workshop Video Proc. and Quality metrics*, 2007. → pages 79, 88

[103] U. Engelke, H.J. Zepernick, and T.M. Kusuma. Wireless imaging quality database. URL http://www.bth.se/tek/rcg.nsf/pages/wiq-db. → pages 80

[104] Final report from the video quality experts group on the validation of objective models of video quality assessment, 2000. URL http://www.vqeg.org. → pages 81

[105] Eric C Larson and Damon M Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006–011006, 2010. → pages 91

[106] S. Winkler. Visual fidelity and perceived quality: Towards comprehensive metrics. In *Proc. SPIE*, volume 4299, pages 114–125, 2001. → pages 92

[107] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64:107–123, 2005. → pages 98, 121

[108] S. Winkler. *Digital video quality*. Wiley, 2005. → pages 110, 112

[109] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. ICCV*, 2003. → pages 119

Appendix A

A Brief Overview of the Human Visual System

The HVS is a part of our *central nervous system* and is responsible for sensing and processing visual information from natural environment. It can be broadly subdivided into two components: a pair of *eyes* that are responsible for capturing the visual information and the *visual pathways* in the brain, which transmit and process the visual signals [108]. The detailed anatomy and physiology of these components are out of the scope of this thesis. Nonetheless, a brief overview, particularly relevant to the interest of this work, is provided below.

A.1 Eyes

Figure A.1 presents a basic diagram of the HVS. As seen in the figure, our eyes serve as the interface between the outside world and the rest of the visual system. A human eye, often compared with a photographic camera, captures the light and focusses the light rays on to the retina - a membrane at the back of the eye.

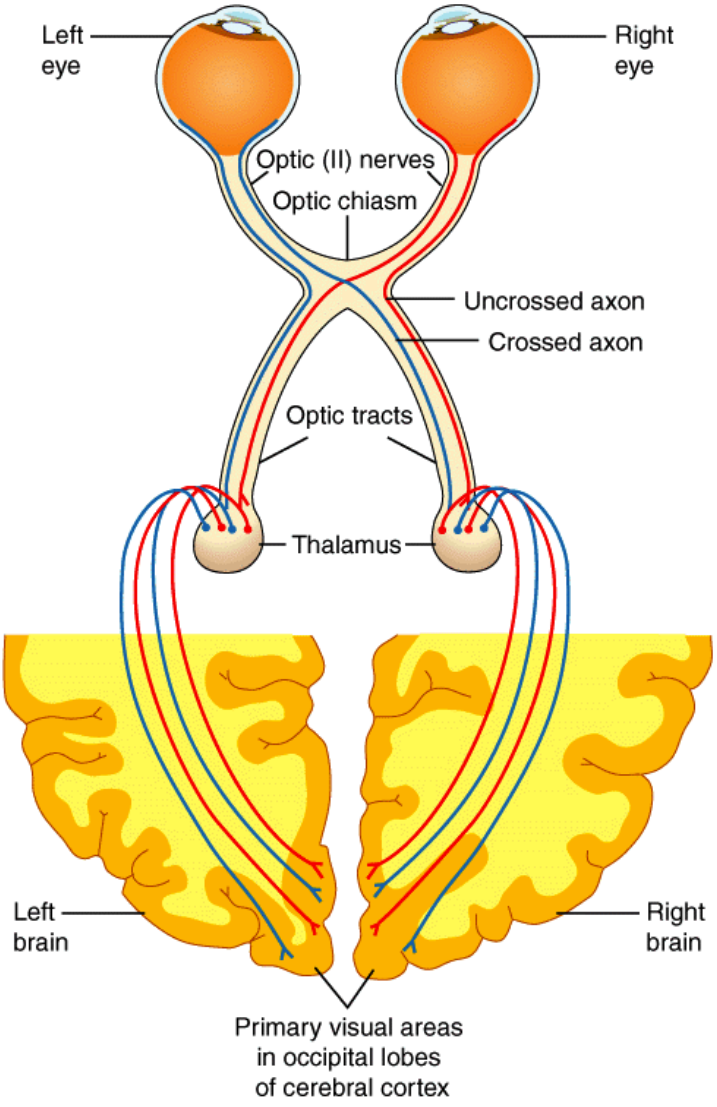


Figure A.1: The human visual system (HVS) [*image from wikipedia.org*]

The retina contains multiple layers of neurons. The first layer contains a set of light-sensitive neurons called the *photoreceptor* cells. There are two types of photoreceptor cells: the *rods* and the *cones*. Rods are sensitive to low light and cones

are sensitive to high light levels. The task of photoreceptors is to convert the optical signal to a form that can be interpreted by the brain. The discretely (not uniformly) sampled signal from the photoreceptors is transferred to the layers of horizontal cells, bipolar cells, amacrine cells and ganglion cells. The ends of the ganglion neurons are connected to the optic nerves which carry the preprocessed signal to the brain.

A.2 Visual pathways

The two optic nerves, carrying the information from the ganglion cells, meet at the optic chiasm (see Figure A.1), where the nerve fibers are rearranged. Half of the fibers from each retina cross to the opposite side and join the temporal fibers of the opposite retina to form the optic tracts. This means that the left retinal image is processed in the right hemisphere of the brain and vice versa. The fibers of each optic tract synapse in the *lateral geniculate nucleus*. The fibers then pass though optic radiation to the *primary visual cortex*.

A.2.1 The primary visual cortex (V1)

The *primary visual cortex* (also known as *striate cortex* or V1) is located in the occipital lobe at the back of the human brain and is responsible for performing all high-level tasks associated with human vision and perception. A large variety of neurons is present in V1. These cells often have selective sensitivity towards certain information; for example, some cells are only sensitive to certain patterns, some cells are sensitive to motion in a particular direction and some other cells are tuned to sense only particular frequencies or color [108].

A class of neurons, called *simple cells*, is of particular importance to the re-

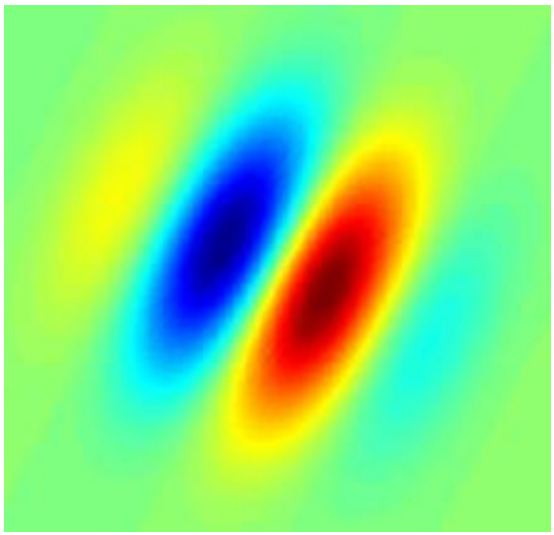


Figure A.2: The receptive field of a simple cell in V1. Blue regions indicate the inhibitory (OFF) regions and red regions mean excitatory (ON) regions. [*image from wikipedia.org*]

searchers dealing with human vision. These cells are primarily responsible for extracting information from oriented edges and gratings. The receptive field of a simple cell is localized, oriented and frequency-selective. It has clear ON and OFF (excitatory and inhibitory) regions (see Figure A.2), indicating that it responds to only those visual stimuli that have a range of spatial frequencies and orientations about its center values.

Appendix B

Additional Applications

B.1 Facial expression recognition

The facial expression dataset [33] involves 2 individuals, each expressing 6 different emotions under 2 lighting setups. The expressions are anger, disgust, fear, joy, sadness and surprise. Expressions such as sadness and joy are quite distinct but others are fairly similar, such as fear and surprise. Under each lighting setup, each individual shows each of the 6 expressions 8 times. The subjects always start with a neutral expression, show an emotion, and return to neutral (see Figure B.1 for sample frames).

Figure B.1: Sample frames from the Facial expression dataset: anger (f1), disgust (f2), fear (f3), joy (f4), sadness (f5) and surprise (f6).

Table B.1: Concatenated dictionary-based classification results are compared with the traditional BoW approach. For true comparison the same detector and descriptors are used both cases.

Condition	Recognition accuracy (%)	
	Dollar et al. [33]	Concat
same subject & lighting	97.9	**100**
same subject, different lighting	89.6	**93.7**
different subject, same lighting	75.0	**91.7**
different subject & lighting	69.8	**72.9**

	f1	f2	f3	f4	f5	f6
f1	**1**	0	0	0	0	0
f2	0	**1**	0	0	0	0
f3	0	.25	**.75**	0	0	0
f4	0	0	0	**1**	0	0
f5	0	0	0	0	**1**	0
f6	0	0	.25	0	0	**.75**

(a)

	f1	f2	f3	f4	f5	f6
f1	**1**	0	0	0	0	0
f2	.25	**.63**	.12	0	0	0
f3	0	0	**.5**	0	0	.5
f4	0	0	0	**1**	0	0
f5	.25	0	0	0	**.75**	0
f6	0	.25	.25	0	0	**.5**

(b)

Figure B.2: Results on the Facial Expression dataset: (a) different subject, same illumination (91.7%) and (b) different subject, different illumination (72.9%)

Recognition results are provided in Table B.1 using the concatenated dictionary-based classification algorithm (*concat*). In order to provide a true comparison with the original work [33], we use the combination of cuboids and the concatenated gradient vector to compute the feature vectors. Comparisons in Table B.1 show that our sparse representation-based algorithm is better than the results reported in [33]. Confusion matrices are presented in Figure B.2.

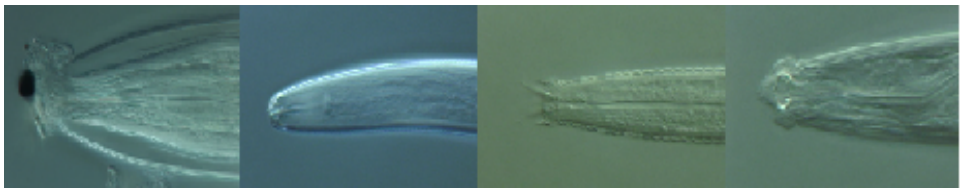


Figure B.3: Sample images from the Nematodes datasets used.

Table B.2: Results on the Nematodes dataset.

Approach	Recognition rate (%)
ℓ_1 optimization [27]	54.0
Compression based [58]	56.0
EFVC	**64.0**

B.2 Biological species classification

The *Nematodes dataset* [58] is a collection of 50 color images (converted to grayscale) of 5 nematode species [58]. Nematodes are a diverse phylum of wormlike animals, with great commercial and medical importance. Nematodes, because of their diversity, are known to be extremely difficult to be classify correctly. Images are downsampled by a factor of 4 to be consistent with the image size and other parameters. We have adopted a leave-one-out scheme for the evaluation of this dataset to allow direct comparisons to the results obtained by the original authors. The classification results obtained using the Error Feature Vector-based Clasification (EFVC) algorithm are presented in Table B.2 . The results show 8% improvement over the state-of-the-art.

Appendix C

LMP Feature Extraction

We define a *local motion pattern* as a distinctive, scale-invariant region that contains significant information about the local variations of the signal along both spatial and temporal dimensions. It was noted in [33], that the extrema points are often located at the regions having spatially distinguishing structure. Consequently, we deduce that the local motion patterns should correspond to the temporal variations in such spatially distinctive regions over a short period of time. Our purpose is to detect the spatially distinctive points and then capture the temporal changes in the neighborhood of those points.

Feature detection: Consider a video sequence $\mathbf{V}(x,y,t)$ consisting of f frames. It is first partitioned into S segments: $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_S]$ (see Figure C.1) such that each segment contains $l = f/S$ consecutive frames. The number of frames in a segment, l, corresponds to the temporal resolution at which $\mathbf{V}$ is analyzed. The smaller the value of l the finer is the resolution. At any given resolution l is required to be large enough to accommodate small movements of the subject but not too large to have any major changes.

Figure C.1: Multiple temporal scales analysis of a video sequence partitioned into 4 and 8 temporal segments for computation of the LMP descriptors.

In order to extract spatially distinguishing structures we employ a 2D keypoint detector and locate keypoints at the first frame of every temporal segment. Say, ρ keypoints are detected in the first frame of a segment $\mathbf{V}_i$. We are interested in observing how the temporal information around each of these ρ keypoints changes over the remaining $(l-1)$ frames. This can be handled by prealigning the subjects (when translation is involved) in all the frames of $\mathbf{V}_i$ w.r.t. a reference point. Then, fixing the coordinate values obtained for the keypoints in the first frame, small video patches of dimension $(\eta \times \eta \times l)$ are extracted around each of the ρ key points, in every $\mathbf{V}_i$, $i = 1,2,..S$.

The prealignment of frames simplifies the process of patch extraction. Often, such prealigned sequences are the output of the tracking procedures used to detect the subject of interest. However it requires a good bounding box and may be difficult in the cases of background clutter or partial occlusion. An alternative to prealignment of the figures is to find the points corresponding to the keypoints detected in the first frame in the next frames, for example, by SIFT feature matching [32]. Note that, prealignment removes all information about a subject's translation, but translation does not contribute much to the recognition process anyway. This

prealignment step is also adopted in [39] and [109].

The descriptor: Every keypoint is associated with a spatio-temporal cube of size $(\eta \times \eta \times l)$. Each cube captures the local space-time changes of the signal and represents a significant motion pattern. The spatio-temporal cubes are extracted in all temporal segments of $\mathbf{V}$. In order to obtain a robust descriptor for each spatio-temporal cube, we first perform 2D Gaussian blurring of each cube in the spatial domain so as to ignore minor variations. This increases robustness of the descriptor against noise and positional uncertainties that are likely to occur from imperfect segmentation or improper alignment, if performed. But the cubes should not be smoothed along the temporal direction so as not to ruin the small temporal variations we are particularly interested in.

Let us denote a blurred cube as $\mathbf{v} \in \mathbb{R}^{\eta \times \eta \times l}$, which is basically a series of l small patches. After removing the mean of $\mathbf{v}$, the second (variance, $\mathbf{M}_2$), third (skewness, $\mathbf{M}_3$) and fourth (kurtosis, $\mathbf{M}_4$) central moments are computed for each pixel along the temporal direction. We define the moment matrix $\mathbf{M}_r$, $r = \{2,3,4\}$ associated with $\mathbf{v}$ as follows:

$$\mathbf{M}_r = [m_{ij}] \quad i,j = 1,2,...\eta \tag{C.1}$$

where

$$m_{ij} = \frac{1}{l} \sum_{t=1}^{l} (v_{ijt})^r \tag{C.2}$$

Here, v_{ijt} is the pixel value at location $\{i,j\}$ of the t-th patch. Each moment matrix $\mathbf{M}_r$, $r = \{2,3,4\}$ is transformed to a vector $\mathbf{m}_r \in \mathbb{R}^{\eta^2}$. The three moment vectors corresponding to three values of r are concatenated on top of each other to form a

Table C.1: Quantitative comparison between Cuboids and LMP

	Cuboids	**LMP**
video size	$101 \times 101 \times 84$	$101 \times 101 \times 84$
temporal scales	3	3
spatial scale	2	2
features extracted	438	474
run time (sec)	16.70	5.08

single vector $\mathbf{m} \in \mathbb{R}^d$ where $d = 3\eta^2$.

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_2 \\ \mathbf{m}_3 \\ \mathbf{m}_4 \end{bmatrix} \tag{C.3}$$

The vector $\mathbf{m}$ is an LMP descriptor. A number of such descriptors that collectively characterize a human action is extracted from each video sequence. The process of computing the LMP descriptors is illustrated in Figure C.2. The advantages of these proposed descriptors are as follows:

- *Computational efficiency* - Assume that the video frames are prealigned. The order of computational complexity of detecting keypoints in an image, using for example, the Harris interest point detector, is $\mathscr{O}(n)$, where n is the number of pixels in the image. For a video sequence divided into S no. of temporal segments, keypoints have to be detected only in S no. of images. If we consider T temporal scales ($T \geq 1$), the complexity is $\mathscr{O}(nC) \sim \mathscr{O}(n)$, where $C = \sum_{j=1}^{T} S_j$ is a small constant and S_j is the number of temporal segments at scale j. Thus the order of complexity of extracting the spatio-temporal cubes is equal to that of the 2D keypoint detector being used. Evidently the

complexity of 2D extrema detection is much lower than the 3D extrema detection used to find the 3D spatio-temporal keypoints in [107, 41, 33]. From Table C.1, we can see that LMP is almost three times as fast as the cuboids.

- *Flexibility* - one can choose from a large pool of 2D keypoint detectors based on the application, data type and quality. Descriptors can be computed for a variety of data types such as silhouettes, blobs and plain grayscale images. Background subtraction is not necessary.

- *Scale invariance* - temporal and spatial scale invariance is easy to achieve by using a multiscale 2D keypoint detector and multiple temporal resolutions.

The demerit of this feature extraction method is the cost of prealignment of the video frames or alternatively, tracking the keypoints in the consecutive frames.

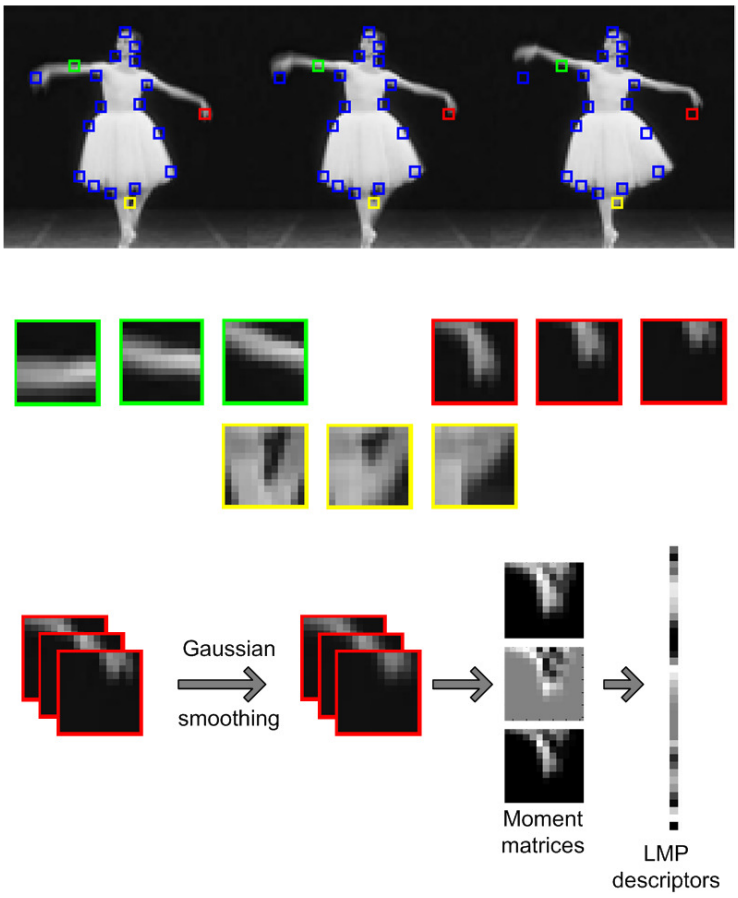


Figure C.2: (a) A temporal segment consisting of three consecutive video frames. The 2D keypoints are identified in the first frame using improved Harris keypoint detector. The positions of the same keypoints are shown in the next two frames. (b) Patches are extracted around each keypoint at each frame. Three space-time cubes associated with the three keypoints (green, red, yellow) are shown. Each cube contains patches extracted from the three frames. (c) Conversion of a cube to an LMP descriptor: Gaussian blurring of the cube is followed by the computation of the 2nd, 3rd and 4th central moments in the temporal dimension and transformation of the three moment matrices into one vector. (This image is best viewed in color.)