SPARSE MODELING APPLIED TO PATIENT IDENTIFICATION FOR SAFETY IN MEDICAL PHYSICS APPLICATIONS

by

Stephanie Lewkowitz

A Thesis Submitted to the Faculty of

The Charles E. Schmidt College of Science

in Partial Fulfillment of the Requirements for the Degree of

Professional Science Master

Florida Atlantic University

Boca Raton, FL

August 2016

SPARSE MODELING APPLIED TO PATIENT IDENTIFICATION FOR SAFETY IN MEDICAL PHYSICS APPLICATIONS

by

Stephanie Lewkowitz

This thesis was prepared under the direction of the candidate's thesis advisor, Dr. George Kalantzis, Department of Physics, and has been approved by the members of her supervisory committee. It was submitted to the faculty of the Charles E. Schmidt College of Science and was accepted in partial fulfillment of the requirements for the degree of Professional Science Master.

SUPERVISORY COMMITTEE:

George Kalantzis, Ph.D.
Thesis Advisor

Theodora Leventouri, Ph.D.

J. S. Faulkner, Ph.D.

Silvia Pella, Ph.D.

Elan Barenholtz, Ph.D.

Luc Wille, Ph.D.
Interim Chair, Department of Physics

Janet Blanks, Ph.D.
Interim Dean, Charles E. Schmidt College of Science

Deborah L. Floyd, Ed.D.
Dean, Graduate College

Date

ACKNOWLEDGEMENTS

First and foremost, my sincere thanks goes to our Medical Physics program director, Dr. Theodora Leventouri, whose passion for seeing her students succeed supports all of us immensely and is hugely appreciated.

Next, my thanks goes to the time and effort of Dr. George Kalantzis, who advised me on my research project and thesis.

Special thanks to my co-authors William Edward Hahn, Daniel Lacombe, and Dr. Elan Barenholtz, as well as members of the Machine Perception and Cognitive Robotics Laboratory.

I am grateful to Dr. Silvia Pella for her generous commitment to helping students in Medical Physics.

ABSTRACT

Author:	Stephanie Lewkowitz
Title:	Sparse Modeling Applied to Patient Identification for Safety in Medical Physics Applications
Institution:	Florida Atlantic University
Thesis Advisor:	Dr. George Kalantzis
Degree:	Professional Science Master
Year:	2016

Every scheduled treatment at a radiation therapy clinic involves a series of safety protocol to ensure the utmost patient care. Despite safety protocol, on a rare occassion an entirely preventable medical event, an accident, may occur. Delivering a treatment plan to the wrong patient is preventable, yet still is a clinically documented error. This research describes a computational method to identify patients with a novel machine learning technique to combat misadministration.The patient identification program stores face and fingerprint data for each patient. New, unlabeled data from those patients are categorized according to the library. The categorization of data by this face-fingerprint detector is accomplished with new machine learning algorithms based on Sparse Modeling that have already begun transforming the foundation of Computer Vision. Previous patient recognition software required special subroutines for faces and different taylored subroutines for fingerprints. In this research, the same exact model is used for both fingerprints and faces, without any additional subroutines and even without adjusting the two hyperparameters. Sparse modeling is a power-

ful tool, already shown utility in the areas of super-resolution, denoising, inpainting, demosaicing, and sub-nyquist sampling, i.e. compressed sensing. Sparse Modeling is possible because natural images are inherrantly sparse in some bases, due to their inherrent structure. This research chooses datasets of face and fingerpint images to test the patient identification model. The model stores the images of each dataset as a basis (library). One image at a time is removed from the library, and is classified by a sparse code in terms of the remaining library. The Locally Comptetive Algorithm, a truly neural inspired Artificial Neural Network, solves the computationally difficult task of finding the sparse code for the test image. The components of the sparse representation vector are summed by ℓ_1 pooling, and correct patient identification is consistently achieved 100% over 1000 trials, when either the face data or fingerprint data are implemented as a classification basis. The algorithm gets 100% classification when faces and fingerprints are concatenated into multimodal datasets. This suggests that 100% patient identification will be achievable in the clinal setting.

DEDICATION

Thank you to my loving, generous, patient parents, and to all of my loving friends and family. Special thanks to William Hahn for his many good ideas and support, which made this work possible.

SPARSE MODELING APPLIED TO PATIENT IDENTIFICATION FOR SAFETY IN MEDICAL PHYSICS APPLICATIONS

Chapter 1

Introduction

Richard Hamming, Bell Labs mathematician and pioneer of computer science error-correcting codes, said in 1995 that he was not interested in the competition, but rather in what humanity and Artificial Intelligence (AI machines) would accomplish together [8]. However, he recognized that it was already faster, more reliable and more cost effective in most cases for machines to analyze medical measurements rather than humans [8]. He proposed future machines would improve diagnoses. "If you have gone to a modern hospital you have seen the invasion of computers: the field of medicine has been very aggressive in using computers to do better in cost reduction, accuracy, and speed. The computers do the billing, scheduling, and record keeping for the mechanics of the hospital. In many hospitals, computers monitor patients in the emergency ward, and sometimes in other places when necessary. The machines do not get bored, respond rapidly, and alert the nurse to act promptly. It is doubtful a nurse could equal the combination of computer and nurse" [8].

In October 2015, Jeremy Howard's Australian based company, Enlitic, raised 10 million dollars for AI in medical imaging [10], bringing to fruition a global scale endeavor to optimize health with big data and surpass medical doctor diagnoses, reminiscient of Hamming's predictions from decades earlier. Enlitic is one of the first companies

with the mission of applying Deep Learning to medicine. Deep learning, layered neural networks, perform unsupervised learning (training on unlabeled data) to build features at multiple scales. "Our team of healthcare executives, physicians, and data science experts are adapting deep learning to medicine, with the goal of bringing patients radically improved diagnostic outcomes. We collaborate closely with medical centers to validate our technology and benchmark our performance against publicly available medical data sets"[10].

In "Deep Learning Human Actions from Video via Sparse Filtering and Locally Competitive Algorithms," research published by the author in July 2015, a multi-layered neural-network with unsupervised feature learning was implemented to correctly categorize handwaving, walking and running videos with 97% accuracy [7]. There has been a lot of momentum in the fields of Computer Vision and Signal Processing leading up to the recent breakthrough of Deep Learning. In large part this is due to breakthroughs in novel methods that accomplish sparse modeling. AI researcher Honglak Lee commented about Deep Learning features, "sparsity regularization during training was necessary to learn oriented edge filters; when this term was removed, the algorithm failed to learn oriented edges" [13]. In Lee's work, a Deep Learning Neural Network trained on thousands of various natural images to produce an output layer consisting of filters, (square patches also called atoms, neurons, and dictionary elements) that compete with each other to represent unique pixel combinations.

First layer features have localized edges with orientation. When this output layer of features is used as the input for a second layer, the network empirically and selectively responded to contours, corners, arcs, and surface boundaries [13], visible on the left of 1.1. Continuing counterclockwise, at the bottom right is a third layer, trained on

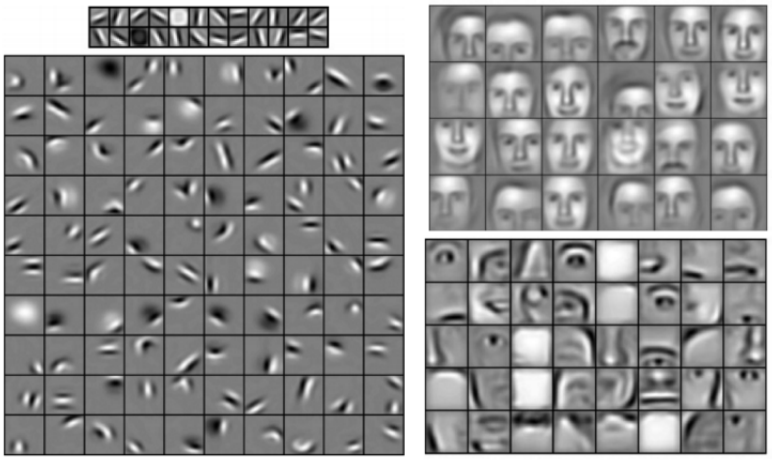

Figure 1.1: Counterclockwise from top left: First and second layers trained on natural images. from bottom right: third and fourth layers trained on faces [13].

a collection of face images built from the first and second layer filters. In the top right is the fourth trained layer, where noticeable elements of faces are evident. The same first two layers (that were trained on all natural images) were used as primitives to build filters of other natural image categories, for example: cars, elephants, and chairs 1.2.

"The dictionary learning approach aims at capturing localized structural information and suppressing noise so that image reconstruction with a sparse representation in terms of dictionary can perform well" [4]. As hinted above, much new research in machine learning works with sparse modeling for classification, as well as a wide variety of other relevant tasks, such as image denoising, reconstruction and compressed sensing, highlighted in the next section.

Given a training set of observed signal samples, learning a dictionary allows for a sparse representation, where a fixed basis may not not yield a sufficiently sparse

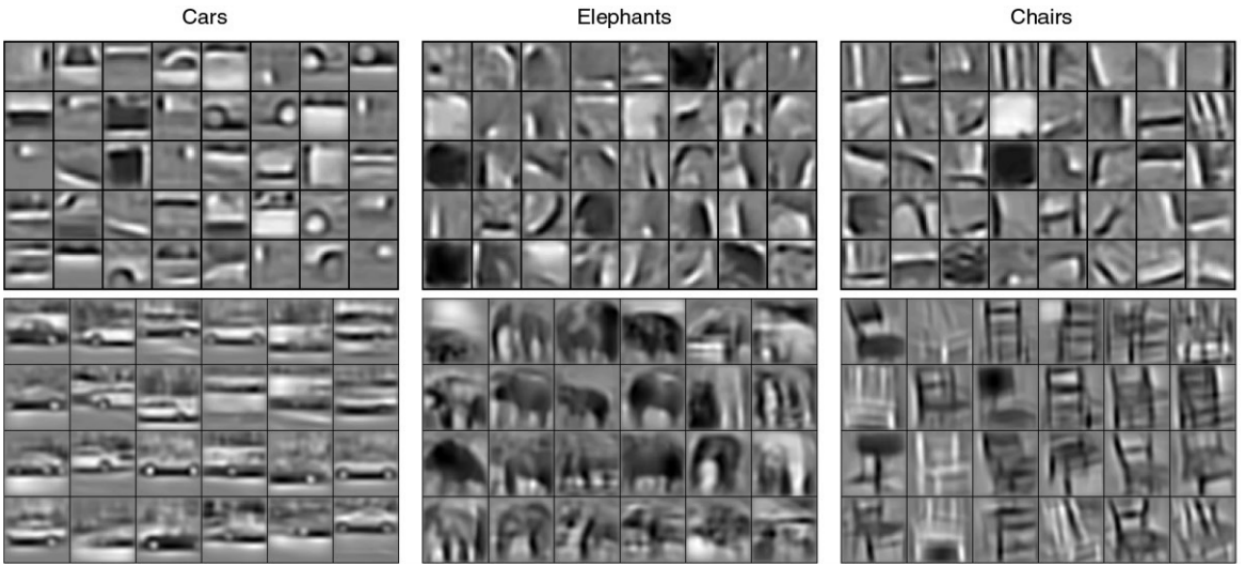

Figure 1.2: Third layer (top) and fourth layer (bottom) filters from different object categories [13].

representation of such signals [21]. However, before dictionary learning was possible, the standard for sparse signal recovery problems employed a known dictionary (basis), such as fourier transform matrices and random matrices, which generated a great amount of theoretical results on sparse recovery [21]. Various other dictionaries, such as wavelets, curvelets and gabors, are now used regularly in image processing research. In this project, the input images themselves are set as the column vectors in the dictionary matrix [26]. We test the model to represent unlabeled images as a linear combination of known images, and show that sparse modeling works as a method for fingerprint recognition and face recognition.

Computer Vision, the growing field of algorithms that sort image and video data, is a valid and valuable tool for Medical Imaging applications. Justification comes from defining a mathematical problem of building sparse representations of images

as vectors, and showing how recent breakthroughs in signal processing give rise to a new class of machine learning neural networks that solve computational tasks with a simple dynamical system, tasks that were formerly thought to be NP-Hard.

1.1 MOTIVATIONS FOR SPARSE MODELING IN MEDICAL PHYSICS

Sparse Modeling is one of the most active areas in Image Processing currently, according to Duke University Professor, Guillermo Sapiro. "The key results in sparse modeling and compressed sensing identify particular conditions on the design matrix and signal sparsity that allow for an accu- rate reconstruction of the signal, as well as optimization algorithms that achieve such reconstruction in a computationally efficient way" [21].

1.1.1 Computer Vision

Computer vision exploits texture, shape, contour and prior knowledge along with contextual information from image sequences to perform tasks of segmentation, machine learning, classification, tracking and reconstruction, revealing new 3D and 4D information [4].

Feature learning In IGRT, Image Guided Radiation Therapy, the patient is under CT scanner during the treatment, to help match the plan to treat the actual location of the patient tumor. Because organs may shift throughout the day, month, and treatment time, placing patients in the correct position is not sufficient information to ensure placement of the tumor or organs at risk. Sparse Modeling with LCA could provide a fast, reliable, robust method to match the stored 3-D contours of patients active in the treatment planning software with the images streaming during the treatment. During this project, we developed a model that runs LCA live from

the camera stream. Hence this model runs LCA and solves the sparse modeling application to identify people by photograph in real time. This project is available on our MPCR Lab github [18].

Segmentation In the medical imaging application of segmentation, strong shape characteristics of biological structures provide pivotal priors that aid segmentation [27]. Recently, a novel shape prior modeling method based on sparse learning theory was proposed, "that was robust to non-Gaussian errors and still preserved individual shape characteristics, even when such characteristics were not statistically significant" [27].

Deep Learning "Deep learning can readily handle a broad spectrum of diseases in the entire body, and all imaging modalities (x-rays, CT scans, etc). Deep learning contextualizes the imaging data by comparing it to large datasets of past images, and by analyzing ancillary clinical data, including clinical reports and laboratory studies"[10].

1.1.2 Compressed Sensing

Native to Medical MRI Imaging, Compressed Sensing is perhaps the most impressive of all the sparse modeling applications. Until recently, sampling a signal at a rate that was at least twice the signal's bandwith, known as Nyquist sampling, was the only reliable way to record the signal [21]. The Shannon-Nyquist sampling theorem states the requirement for a minimum number of samples in order to perfectly capture a bandlimited signal [6]. In practice, Nyquist sampling produces a tremendous number of measurements and requires a compression step in order to efficiently store or transmit the data [21]. The compression step represents a signal with a known

basis (e.g. Fourier, wavelets, etc.,) and sets most coefficients to zero, keeping only a small fraction. Throwing most of the coefficients away compresses the signal but at the cost of increasing the error. In the field of signal processing there was a search for means to combine the compression step with the acquisition step [21]. This would vastly speed up acquisition by reducing the necessary measurements, and reduce error by acquiring a signal in a more suitable basis.

Compressed Sensing is a way to directly sense the data in a compressed form i.e., at a lower sampling rate, enabling a large reduction in the sampling and computation costs for sensing signals that have a sparse representation; Candes, Romberg, Tao and Donoho showed that a finite-dimensional signal having a sparse or compressible representation can be recovered from a small set of linear, nonadaptive measurements [6]. In Compressed Sensing, measurements of a signal are done by a sensing matrix. Because the image will later be reconstructed, the sensing matrix must preserve the pair-wise distance of the signal [6]. "The key idea behind Compressed Sensing is that the majority of real-life signals, such as images, audio, or video, can be well approximated by sparse vectors, given some appropriate basis, and that exploiting the sparse signal structure can dramatically reduce the signal acquisition cost; moreover, accurate signal reconstruction can be achieved in a computationally efficient way, by using sparse optimization methods" [21].

Facility Demographics

State

State	N	State	N	State	N	State	N	State	N
AK	1	HI	1	ME	2	NJ	14	SD	1
AL	11	IA	8	MI	19	NM	4	TN	9
AR	6	ID	1	MN	10	NV	2	TX	44
AZ	20	IL	24	MO	13	NY	37	UT	5
CA	50	IN	18	MS	5	OH	16	VA	14
CO	10	KS	7	MT	2	OK	8	VT	0
CT	4	KY	10	NC	19	OR	6	WA	17
DE	1	LA	6	ND	2	PA	39	WI	16
FL	31	MA	18	NE	5	RI	0	WV	4
GA	15	MD/DC	16	NH	1	SC	6	WY	1

Figure 1.3: Distribution of facilities by state, 2014 Radiation Therapy Staffing and Workplace Survey, American Society of Radiologic Technologists[1].

1.2 MOTIVATIONS FOR PATIENT IDENTIFICATION PROGRAM

The motivations for exploring and validating the utility of sparse modeling for face recognition and fingerprint recogntion are firstly that we can check the solution with our own eyes, and secondly the genuine need for such a computer program in the clinical setting.

The following satistics and four figures are from the 2014 Radiation Therapy Staffing and Workplace Survey conducted by the American Society of Radiologic Technologists. The survey was emailed to 3,524 managers of U.S. radiation therapy facilities in February, 2014. There were 654 completed submissions, with a response rate of 18.6%. The sample size of 654 yields a margin of error of a maximum 3.8% (at the 95% confidence interval) for overall percentages [1].

Radiation Therapy clinics are distributed throughout the United States, and of course, throughout the world. Many types of employees are staffed at the clinics: Medical Physicians, Medical Physicicst, Medical Dosimetrists, Radiation Therapists, Engi-

neers, IT and vital clinical staff are all important to the collective function: diagnose, plan and deliver treatments to patients fighting cancer. The most frequent services provided by facilities include Intensity-modulated radiation therapy, IMRT (95.2%) , 3D-Conformal radiation therapy (92.9%), and Image-guided radiation therapy, IGRT, (88.9%). All living cells are affected by ionizing radiation.

Safety is especially important in clinical Medical Physics, where even little mistakes have the potential to cause great harm to patients. Because people are human and do occassionally make mistakes, there are recorded events in the clinical setting that may be averted in the future by monitoring the identify of the patient with software.

A medical misadministration can occur when a patient is administered a radiopharmaceutical or radiobiological substance. A medical misadministration can also occur during the performance of external beam treatments or brachytherapy procedures [9].

During the period of 1/1/2001 through 12/31/2009, According to the State of New York Department of Health Bureau of Environmental Radiation Protection, 230 medical therapy accelerator misadministrations were reported. The wrong patient was treated in 19% of these events, due to failure to ensure the correct patient was in the room [12]. The main goal of this research is to build and explore a patient recognition algorithm that may one day help solve the problem of misadministration, by equipping clinical staff with software that visually checks the patient in the treatment room with the patient plan active in the treatment planning software.

The patient identification program stores face and fingerprint data for each patient. New, unlabeled data from those patients are categorized according to the library.

Number of patients receiving treatment per day

	Frequency	Valid Percent	Cumulative Percent
1 to 12	30	5.2%	5.2%
13 to 25	139	24.3%	29.5%
26 to 38	126	22.0%	51.5%
39 to 65	150	26.2%	77.7%
66 to 140	103	18.0%	95.6%
141 and over	25	4.4%	100.0%
Total	573	100.0%	
Mean	52.7 (SD= 54.8)		
Percentiles	5th=13, 25th=25, 50th=38, 75=63, 95th=139		

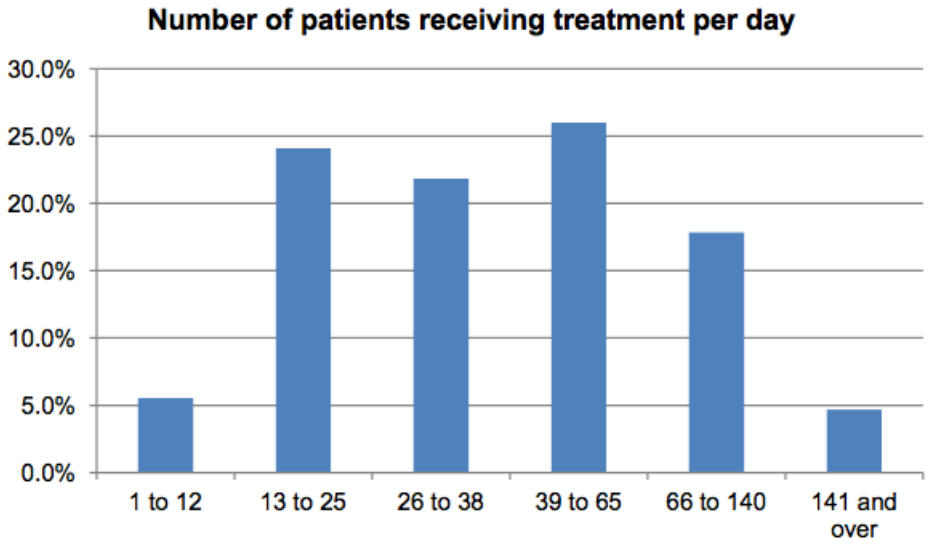

Figure 1.4: Average patients treated daily, 2014 Radiation Therapy Staffing and Workplace Survey, American Society of Radiologic Technologists[1].

The field of Computer Vision has experienced recent breakthroughs in regards to machine learning algorithms, inspiration for this face-fingerprint detector are based on a dynamical system whose behavior naturally evolves to the solution of the sparse modeling problem.

How many hours per day does your facility routinely schedule only one radiation therapist per linear accelerator?

	Frequency	Valid Percent	Cumulative Percent
0 hours	406	73.0%	73.0%
Less than 1 hour	12	2.2%	75.2%
1 hour	38	6.8%	82.0%
Between 1 and 2 hours	10	1.8%	83.8%
2 to 4 hours	47	8.5%	92.3%
5 to 8 hours	34	6.1%	98.4%
More than 8 hours	9	1.6%	100.0%
Total	556	100.0%	
Mean	1.0 hour (SD=2.8 hours)		
Percentiles	5th=-, 25th=-, 50th=6 minutes, 75th=43 minutes, 95th=4 hours and 46 minutes		

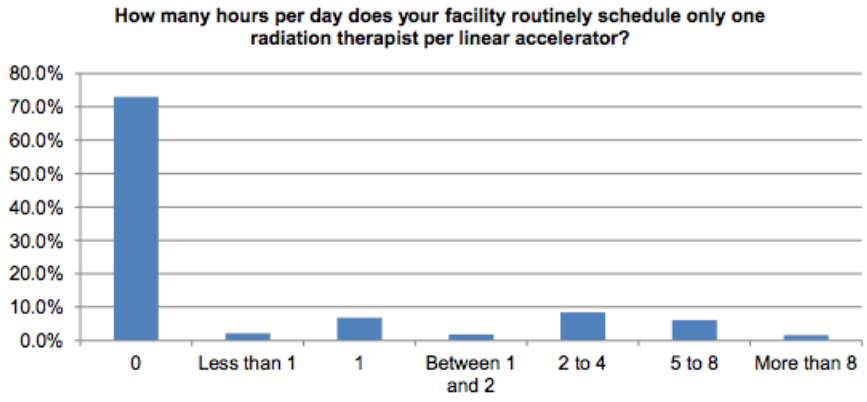

Figure 1.5: Hours per day with one RT per accelerator, Radiation Therapy Staffing and Workplace Survey, ASRT 2014 [1].

Table 2.1 Standard approaches in pattern recognition.

Measured features	Transformation of features	Structural features
Amplitude	Polynomials	Peaks
Bias	Harmonic analysis	Derivatives
Duration	Fourier transform	Lines
Phase	Wavelet transform	Edges
Energy	Haar transform	LPC coefficients
Moments	Karhunen-Loeve transform	Parametric models
Singular values		
Karhunen-Loeve eigenvalues		
Feature selection	**Classifiers**	**Clustering methods**
Discriminant analysis	Euclidian distance	Isodata algorithm
Chernoff bound	Mahalanobis distance	Fisher's linear discriminant
Bhattacharya	Linear discriminant functions	
Divergence	Bayesian linear classifier	Parsing
Exhaustive search	Maximum likelihood	
Dynamic programming	Production rules	
	Density functions	
	Parzen estimator	
	k-NN algorithm	
	Histogram	

Figure 1.6: Table of modern signal processing techniques for accomplishing pattern recognition [17].

1.3 RELEVANT BACKGROUND INFORMATION

1.3.1 Sparse Modeling

Natural images tend to be highly structured, with strong correlations in neighboring pixel values as well as repetition of spatial and temporal patterns within and across images. Sparse coding is a method of representing data based on a linear combination of a few dictionary elements [15]. In recent years, much multi-disciplinary research has been conducted on sparse models and their applications [15]. Sparse approximation is a difficult non-convex optimization problem that is at the center of much research in mathematics and signal processing [24].

Sparse modeling has in the last decade been shown to be useful as a foundation for computer vision and image analysis. Sparse modeling has shown utility in the areas of super-resolution, denoising, inpainting, demosaicing, and sub-nyquist sampling, i.e. compressed sensing. As early as 2010, automatic face recognition posed a significant challenge to computer vision [26].

Biological vision itself may depend on sparse methods to compress and store input, as well as attach semantic information to inherent statistical structure of natural scenes [20]. Physiological evidence suggests that visual cortex compresses the visual input on the basis of spatial and orientation-tuned filters, such as gabor wavelets, that are selective to specific orientations, sizes and directions of motion [11].

1.3.2 Locally Competitive Algorithms

LCA is a class of algorithms based on thresholds and local competition to solve a family of sparse approximation problems [24]. Locally competitive algorithms (LCA)

represent a neurologically-inspired class of nonlinear dynamic neural networks that achieve sparse modeling via leaky integrators interacting through local, nonlinear competition between units. LCA is a parallel dynamical system for computing sparse representations of data. LCAs have been shown to provide greater stability than other methods in response to input perturbations.

A biological interpretation of LCA is a network of nodes. When the system is presented with an input image, the collection of nodes evolve according to fixed dynamics [24]. Inputs cause the membrane potential of a node to charge up like a leaky integrator. The change in membrane potential is proportional to how much the input resembles the appropriate dictionary element and the amount of inhibition received from other nodes [24]. When the membrane potential charges sufficiently and rises over a threshold, an action potential is produced for extracellular signaling. Stronger nodes prevent weaker nodes from becoming active, resembling the lateral-inhibitory behavior observed in many retinal cells [24].

1.3.3 LCA in Hardware

Natural signals contain few active features, represented by a small number of non-zero code elements, and can be well-approximated by a subset of elements from an overcomplete dictionary [24]. This method of sparse approximation is the foundation for many modern sensing and signal processing applications [24]; this is mainly due to its ability to efficiently and accurately represent a signal, as digital systems waste time and energy digitizing information that is eventually discarded during compression [24].

Implementable on reconfigurable analog hardware, LCAs could provide the ultra-efficient high-performance computing techniques needed for future computer vision

applications. LCAs can be implemented using a parallel network of simple elements that match well with parallel analog computational architectures [24]. LCA may offer in the future an ultra-low power physical solution to achieving L1-constrained least squares optimization.

1.4 NEW CONTRIBUTIONS TO THE FIELD

It is the purpose of this thesis to demonstrate the utility of a non-linear signal processing model, such as sparse modeling, for image recognition. This research presents the first application of Locally Competitive Algorithms to solving a Sparse Modeling optimization problem for face recogntion and fingerprint recognition, without any adaptation to the model. The sparse representation is measured in three different ways, experimentally validating the utility of L1 Pooling method for classification with sparse modeling natural images.

Chapter 2

Math Framework

"Pattern recognition tasks require the conversion of pattern in features describing the collected sensor data in a compact form. Ideally, this should pertain only to relevant information. Feature selection methods can be either classical methods (statistic or of syntactic nature) or biologically oriented (neural or genetic algorithm based) methods. Feature extraction and selection in pattern recognition are based on finding mathematical methods for reducing dimensionality of pattern representation. A lower-dimensional representation based on pattern descriptors is a so-called feature. It plays a crucial role in determining the separating properties of pattern classes. The choice of features, attributes, or measurements has an important influence on: (1) accuracy of classification, (2) time needed for classification, (3) number of examples needed for learning, and (4) cost of performing classification" [17].

Techniques from sparse signal representation are beginning to see significant impact in computer vision. A simple sparse model consists of a dictionary D and coefficient vector α. We can think of D as a set of fixed signals from which we can build linear combinations in an attempt to match an input signal. Much work has been done in the areas of dictionary learning (for example, see Hahn, Lewkowitz, 2015) The data itself can also be used to form a dictionary. The columns of D form the signals

from which linear combinations attempt to reproduce the signal, coefficient vector α gives the relative strength of each of the dictionary elements contribution in the reconstruction. Matrix multiplication $D\alpha$ provides the approximation of input signal x. The solution to $x = D\alpha$ is under-determined without an additional constraint. One way to choose amongst the infinite possible solutions of α is to impose a prior. Traditionally, ℓ_2 priors defined below have been used to measure the so-called energy, or ℓ_2 norm, of α . We can seek to optimize the vector α such that it has low energy, but it has been shown that this yeilds a poor reconstruction. This is analogous to a least-squares technique.

2.1 PRELIMINARY METHODS

For much of its history, signal processing has focused on signals produced by physical systems. Many natural and man-made systems can be modeled as linear. Thus, it is natural to consider signal models that complement this kind of linear structure. This notion has been incorporated into modern signal processing by modeling signals as vectors living in an appropriate vector space. This captures the linear structure that we often desire, namely that if we add two signals together then we obtain a new, physically meaningful signal. Moreover, vector spaces allow us to apply intuitions and tools from geometry in R3, such as lengths, distances, and angles, to describe and compare signals of interest. This is useful even when our signals live in high-dimensional or infinite-dimensional spaces. All signals in this study are discrete with finite domain [6].

2.1.1 Inner product

The inner product is a preliminary method to measure the angle between two vectors. Given two vectors $\mathbf{x}$ and $\mathbf{y}$ in Hilbert Space, then the inner product $\mathbf{x} \cdot \mathbf{y}$ is given by the sum

$$\sum_i x_i y_i \;=\; |\mathbf{x}||\mathbf{y}| \cos\theta$$

where the inverse cosine of the ratio between this sum and the product of the magnitudes $\frac{\sum_i x_i y_i}{|\mathbf{x}||\mathbf{y}|}$ gives an angle between the vectors. The closer the correlation between the vectors, the closer the angle is to 1. The angle closest to 1 is used for testing classification.

2.1.2 Euclidean Norm

The Euclidean norm, or L2 norm, is found by taking the difference between two vectors, squaring each component, adding these and taking the square root, such as:

$$\text{L2 Norm } (\mathbf{x}, \mathbf{y}) = \left(\sum_i (x_i - y_i)^2 \right)^{\frac{1}{2}}$$

To test classification capabilities, the smallest norm would indicate the closest match between vectors.

2.2 SPARSE MODEL PROBLEM

For a dictionary with n classes and k images per class,[26]

$$D \doteq \begin{bmatrix} D_1, & D_2, & \ldots, & D_n \end{bmatrix} \;=\; \begin{bmatrix} \phi_{1,1}, & \phi_{1,2}, & \ldots, & \phi_{n,k} \end{bmatrix}$$

Dictionaries are used in signal processing for image representation and compression. Dictionary elements, the column vectors of a dictionary matrix, may also be called

atoms, neurons, features, filters, codebooks, and receptive fields. Traditional dictionaries include the Fourier and wavelet basis. Fourier basis expansion provides for the compression of band-limited signals (resolution-limited images). A signal in $\mathbb{R}^n$ is said to be sparse if in a given basis expansion most of the coefficients are zero. For example, a signal is sparse in the Fourier domain if there are a small number of nonzero coefficients in the signal's Fourier expansion. Natural images are sparse in the wavelet domain and thus are compressible. JPEG 2000 takes advantage of this wavelet basis to reduce the number of bits required to send and store images.

2.2.1 Sparse Approximation

A sparse model consists of a dictionary

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} ||\mathbf{a}||_0 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\mathbf{a} \tag{2.1}$$

Dictionary learning is the task of finding the unique $\mathbf{D}$ that yields the sparsest representation for each set of signals. The dictionary is constructed to have m vectors $\{\phi_m\}$ that span the space $\mathbb{R}^n$. Choosing $m > n$ establishes an overcomplete dictionary. The sparsity problem is to minimize the error ε with a sparsity constraint on $\mathbf{a}$

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} ||\mathbf{a}||_0 \quad \text{s.t.} \quad \mathbf{x} - \mathbf{D}\mathbf{a} = \varepsilon$$

Where $||\mathbf{a}||_0$ is the ℓ_0 pseudo-norm, which counts the number of non-zero basis coefficients. The dictionary matrix is comprised of m feature vectors denoted as ϕ_i,

$$\mathbf{x} = \sum_{i=1}^{m} a_i \phi_i \ + \ \varepsilon$$

where $\mathbf{x}$ is represented as a linear combination of dictionary elements plus residual error.

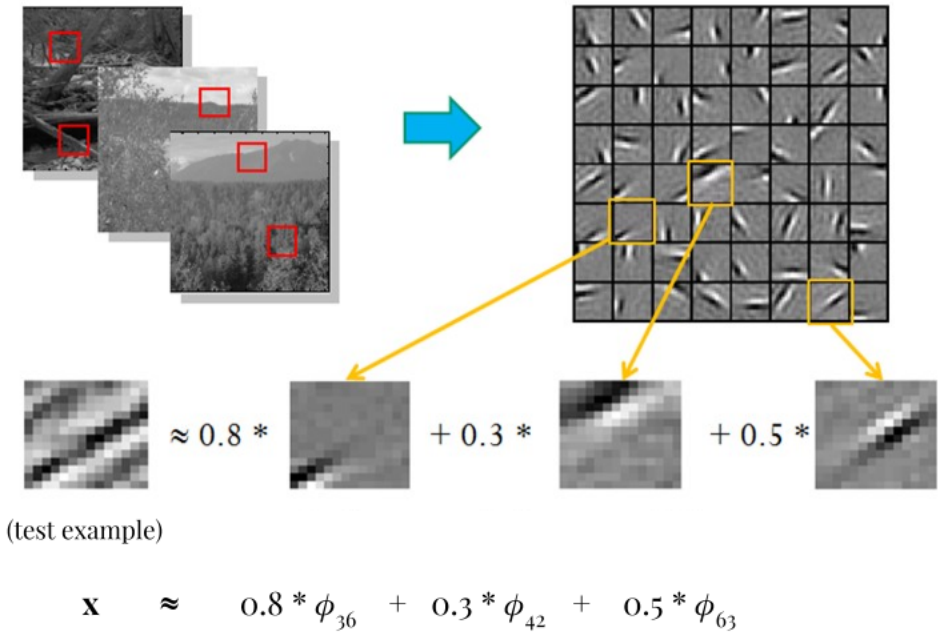

Figure 2.1: Illustration of sparse coding a test patch with natural images [19].

The coefficients of the sparse vector **a** can be described with the following,

$$a_i(x) = \sum_{k=i}^{N} c_{ik}\ \phi_k(x)$$

as a linear combination of dictionary atoms (the column vectors) with coeficient c.

Natural image are structured. Of all possible images, natural images make up a small fraction. Consider the space of binary images (black and white) 100×100 square, 10,000 pixel images. The total number of possible arrangements is $2^{10,000}$, approximately $1.995\times\ 10^{3010}$, a number with 3011 decimal digits. For comparison, the estimated number of atoms in the universe is about $1\times\ 10^{80}$. The number of possible images is orders of magnitude greater. To consider a gray scale, the number of pos-

sible images would be about $2.5 \times 10^{24,082}$, an inconceivable count of possibilities for even such a small image. Most combinations of pixels will be random, or noise. Only a small subset of the total combination would depict a natural scene (mountains, ocean, people, buildings, etc.) Due to the structure inherent in natural images, there exists an optimal basis for sparse representations of natural images. In , a test patch $\mathbf{x}$ is represented as a sparse linear combination of dictionary elements ϕ_m. In this case, the sparse feature vector $\mathbf{a}$ for this test patch can be represented as

$$\mathbf{a} \doteq \left[\phi_1, \ldots, \phi_{64}\right] \doteq \left[\, 0,\ 0,\ .\,.\,0,\ 0,\ \mathbf{0.8},\ 0,\ 0,\ \ldots, 0,\ 0,\ \mathbf{0.3},\ 0,\ 0,\ \ldots, 0,\ 0,\ , \mathbf{0.5},\ 0\right]$$

In optimal sparse approximation, we seek the coefficients having the fewest number of nonzero entries by solving the minimization problem. In this research experiment, given a fixed dictonary D, the challenge is to find a sparse representation of a test image as a linear combination of basis elements, and attempt to classify the identity of the test image from the sparse representation.

"The key idea is a judicious choice of dictionary: representing the test signal as a sparse linear combination of the training signals themselves. We will first see how this approach leads to simple and surprisingly effective solutions to face recognition [26].

Our approach to face recognition assumes access to well-aligned training images of each subject sitting in different poses. Face recognition in the presence of varying illumination and occlusion can be treated as the search for a certain sparse coefficient vector α_0, in the presence of a certain sparse error e. The number of unknowns in (3) exceeds the number of observations, and we cannot directly solve for α_0 [26].

Solving the constrained optimization problem

$$\min_{\alpha} \ ||x - D\alpha||_2^2 + ||\alpha||_0 \tag{2.2}$$

is generally intractable. It can be solved using greedy algorithms such as matching pursuit, or relaxation methods such as basis pursuit, that replace the L^0 norm with the L^1 norm.

The process of choosing a good subset of dictionary elements along with the corresponding coefficients to represent a signal is known as sparse approximation [24]. For a given signal the goal is to select a sparse set of atoms from a given dictionary that well represent the input signal. This subset selection problem is NP-hard, and one main technique is relax the ℓ_0 pseudo-norm constraint by replacing it with the ℓ_1 norm, reducing the problem to convex programming [3]. In the construction process the input signals are represented as combinations of the adapted dictionary vectors. When the dictionary is overcomplete, there are an infinite number of possible representations.

Given an N-dimensional stimulus $s \in \mathbb{R}^N$ we seek a representation in terms of a dictionary $\mathcal{D}$ composed of M vectors $\{\phi_m\}$ that span the space $\mathbb{R}^N$. When the dictionary is overcomplete ($M > N$), there are an infinite number of ways to choose coefficients $\{a_m\}$.

In 2.4 we see a visualization of a sparse subspace. We see the union of all sparse planes in 3D. In other words, for the plane to be considered 2-sparse, it must have one zero component in one of the dimensions. These are the x-y, y-z, and x-z planes.

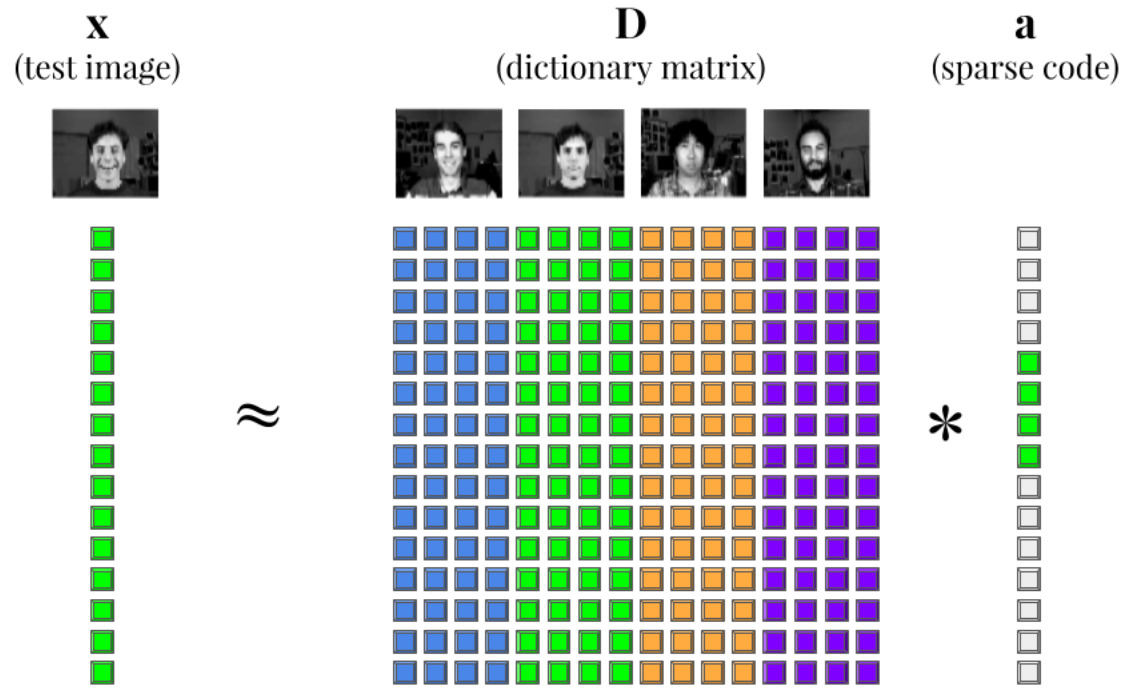

Figure 2.2: illustration of sparse model task, with optimal sparse **a** highlighted green.

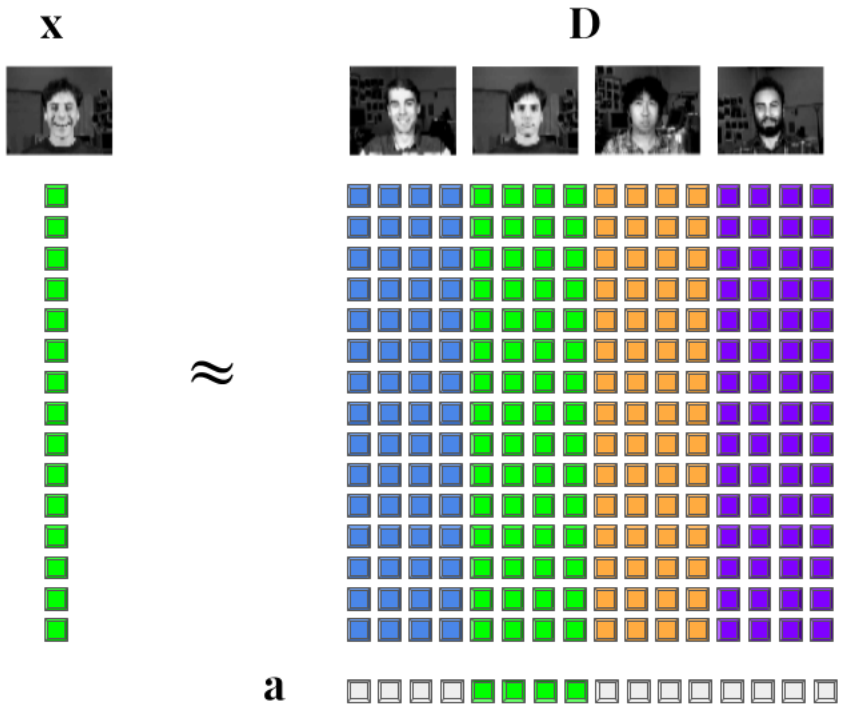

Figure 2.3: illustration of encoding column vectors into the spasre vector.

Each of these planes is sparse (zero) in one dimension.

where the ℓ^0 "pseudo-norm" denotes the support of a vector, the number of nonzero elements of $a = [a_1, a_2, \ldots, a_M]$

Depicted in 2.5, we define the ℓ^p norm of the vector to be

$$||x||_p = \left(\sum_m |x_m|^p \right)^{(1/p)} \tag{2.3}$$

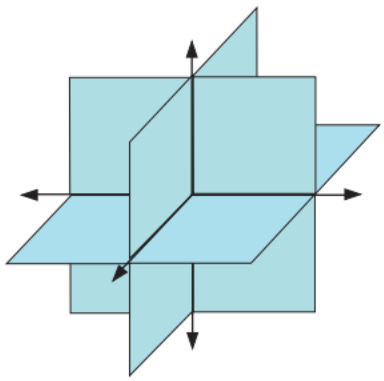

Figure 2.4: Union of subspaces defined by $\sum_2 R^3$, i.e., the set of all 2-sparse signals in R^3[6].

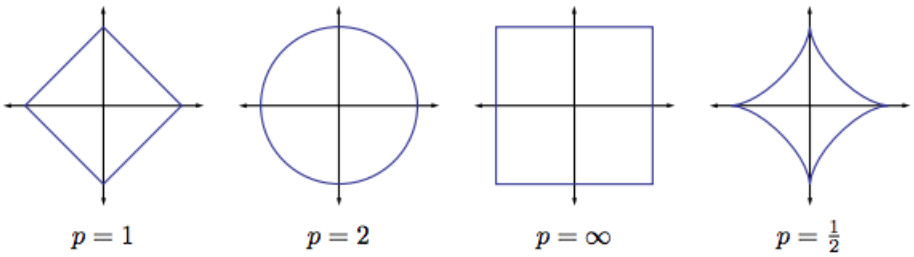

Figure 2.5: unit spheres in R for the ' ℓ_p norms with p = 1, 2, ∞, and for the ℓ_p quasinorm with p = 1/2. [6].

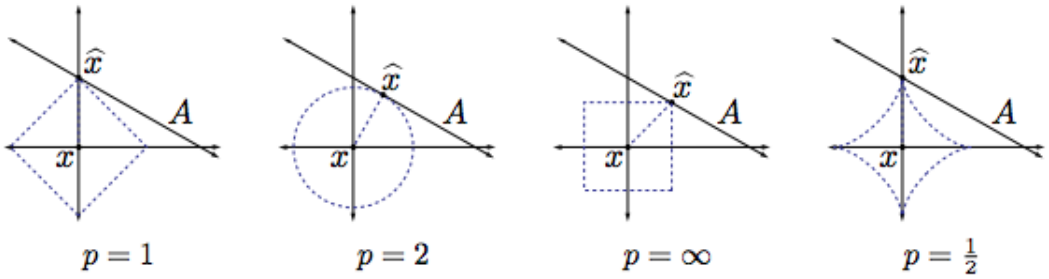

Figure 2.6: Best approximation of a point in R_2 by a one-dimensional subspace using ℓ_p norms with p = 1, 2, ∞, and for the ℓ_p quasinorm with p = 1/2. [6].

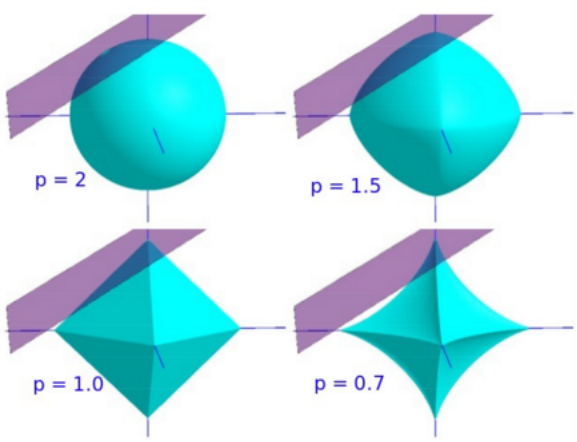

Figure 2.7: Approximation of a point in R_3 by a two-dimensional subspace using ℓ_p norm with different p-norms [5].

2.3 ATOMIC DECOMPOSITION

Atomic decomposition, the method of solving for the sparse representation, is accomplished here with a relaxation technique (basis pursuit).

For a given signal the goal is to select a sparse set atoms from a given dictionary that well represent the input signal. Because this subset selection problem is NP-hard as just mentioned, and thus we must either resort to greedy heuristics or relax the ℓ_0 pseudo-norm constraint by replacing it with the ℓ_1 norm, reducing the problem to convex programing [3]

In optimal sparse approximation, the goal is to maximize the total number of the a_i coefficients set to zero. This task is equivalent to minimizing the ℓ_0 norm:

$$\min_a \quad ||\mathbf{a}||_0 \quad \text{subject to} \quad x = \sum_{i=1}^{m} a_i \phi_i \tag{2.4}$$

This combinatorial optimization problem is NP-hard. It has been shown that convex relaxation of the ℓ_0 norm to the ℓ_1 norm essentially accomplishes the same goal and is practical to implement. The task becomes:

$$\min_{\mathbf{a} \in \mathbb{R}^m} ||\mathbf{D}\mathbf{a} - \mathbf{x}||_2^2 + \lambda ||\mathbf{a}||_1 \tag{2.5}$$

optimized for sparseness by minimizing the ℓ_1 norm.

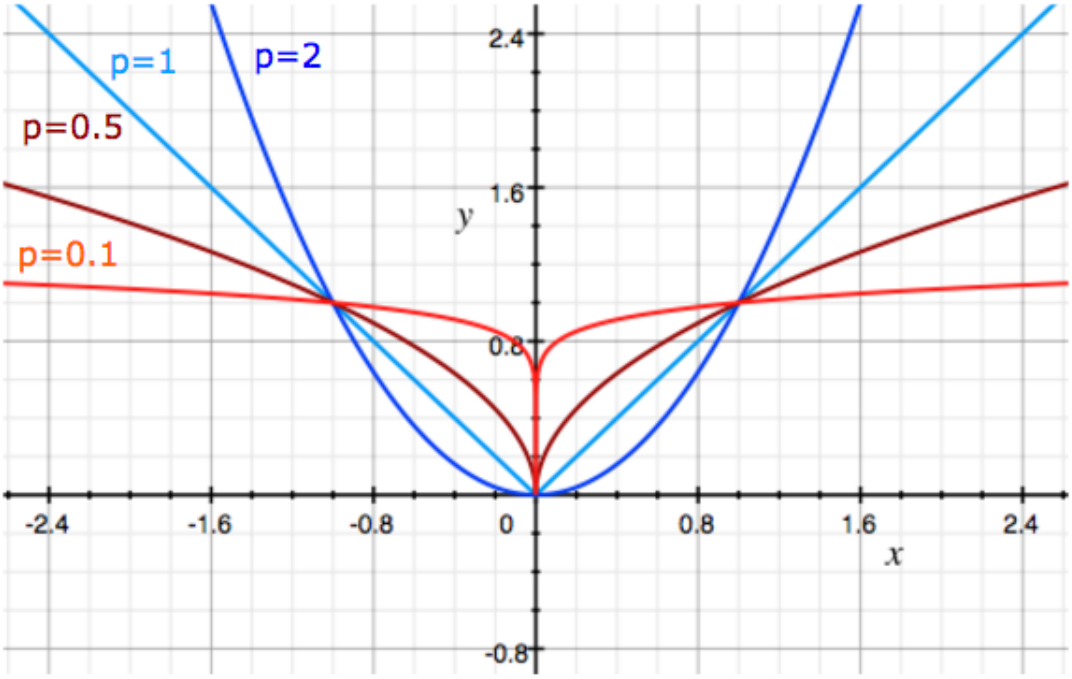

Figure 2.8: p-norm penalty functions.

2.4 DYNAMICAL SYSTEM FOR SPARSE RECOVERY

There are simple systems of nonlinear differential equations that settle to the solution of

$$\min_{\mathbf{a}} \ \lambda||\mathbf{a}||_1 \ + \ \frac{1}{2}||\mathbf{Da}-\mathbf{x}||_2^2 \tag{2.6}$$

The LCA is a neurologically inspired system which settles to the solution of the above.[22] Developed to represent the "equation of motion" for a slice through a cortical column, LCAs are systems of nonlinear differential equations that settle to a minima of a given ℓ_1 regularized least squares optimization problem.

LCAs are defined by a system of differential equations where the initial conditions define an optimization problem and the dynamics converge to a sparse decomposition

of the input vector. LCAs have been shown to provide great stability in response to input perturbations.

Here we use LCA for atomic decomposition: given an input signal x and a pseudo-overcomplete dictionary $\mathbf{D}$, the LCA returns a sparse vector α such that $\mathbf{D}\alpha \approx x$.

The three main components of LCA are leaky integration, nonlinear activation and inhibition/excitation networks [23]. Input to the LCA equations are a stimulus pattern and dictionary, and the output is a sparse code, i.e. a vector of dictionary coefficients. This set of coefficients can now be used as a feature vector for machine learning and classification.

The LCA model approximates the input $\mathbf{x}$ as a linear combination of receptive fields (dictionary elements, or feature columns). $\mathbf{x}$ is approximated as $\hat{\mathbf{x}}$, the product of a sparse vector $\mathbf{a}$ multiplied by receptive fields,

$$\hat{\mathbf{x}}(t) = \sum_m a_m(t)\phi_m \tag{2.7}$$

The sparse coefficient vector $\mathbf{a}$ is determined by solving the LCA differential equations.

$$\dot{v}_m(t) = \frac{1}{\tau}\left[b_m(t) - v_m(t) - \sum_{n \neq m} G_{m,n}a_n(t)\right] \tag{2.8}$$

A nonlinear threshold function is needed for LCA to convert a membrane potential $\mathbf{v}$ into a firing rate,

$$a_m = T_m(\mathbf{v}) = \begin{cases} 0, & \mathbf{v} \leq \lambda \\ \\ \mathbf{v}, & \mathbf{v} > \lambda \end{cases} \tag{2.9}$$

$b_m(t)$ represents the similarity between the m^{th} receptive field and input stimulus, measured with an inner product,

$$b_m(t) = \langle \phi_m, \mathbf{x}(t) \rangle \tag{2.10}$$

$G_{m,n}$ measures the similarity between any two receptive fields ϕ_m and ϕ_n with an inner product,

$$G_{m,n} = \langle \phi_m, \phi_n \rangle \tag{2.11}$$

Note that the receptive fields, ϕ_m, are the columns of the dictionary $\mathbf{D}$, i.e. the feature vectors. Inhibition allows stronger nodes to prevent weaker nodes from becoming active, which results in a sparse solution. Specifically, the inhibition signal from the active node m to any other node n is proportional to the activity level a_m and to the inner product between the node receptive fields.

Originally inspired by visual cortex, LCA has been shown to provide stable atomic decompositions with dynamic inputs.

When the internal state of a node becomes significantly large, the node becomes active and produces an output signal to represent the stimulus and inhibit other nodes [24]. Nodes in a population continually compete with neighboring units using lateral inhibition to calculate coefficients representing an input in an overcomplete dictionary [24]. LCA dynamical system is stable to guarantee that a physical implementation is well behaved [24].

The internal state signal in each node is calculated as a function of said matching

signal received at said node and weighted outputs of all other nodes [24]. Excitatory input current is proportional to how well the image matches with the node's receptive field [24]. This output coefficient is the result of an activation function applied to the membrane potential [24]. The nodes best matching the stimulus will have internal state variables that charge at the fastest rates and become active soonest [24]. This can be modeled as a network of coupled leaky integrators. An imaging system using VLSI to implement LCAs as a data collection front end has the potential to be extremely fast and energy efficient [24]. In LCA, time and energy resources would only be spent digitizing coefficients that are a critical component in the signal representation [24]. Positive and negative coefficients are allowed, but rectified systems could use two physical units to implement one LCA node [24]. LCA can implimented in hardware, for example, on FPGAs or memristor arrays [25].

Chapter 3

Experimental Setup

To demonstrate the robust utility of the LCA model, we measure the sparsity of a signal with three different methods, and test on two spearate datasets, without adapting the model or model parameters. All of the experiments are run with Matlab. The experiment starts with two distinct datasets; one of faces, with eight photos of ten different people in four main poses, the other of fingerprint scans, with eight scans of a finger for each ten patients. Only one dataset is used at a time. The images themselves are stacked into vectors to form the Dictionary. Before the experiment is run, the images are randomized, and one image of each patient category is omited from the dictionary for testing. Hence, there are 10 test images and 70 training images per trial. The model employs LCA to find the sparse reconstruction of that image as a linear combination of the remaining images. With ℓ_1 pooling, the activation levels

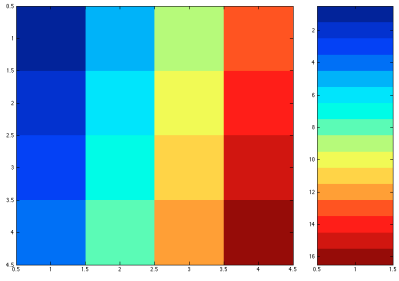

Figure 3.1: left: representation of image as matrix, right: image as column vector

for each patient were assessed during reconstruction of an image.

Each gray-scale image is first converted into a double precision matrix. The matrix is then stacked as a vector. For example, the matrix

$$C = \begin{bmatrix} 0 & 4 & 8 & 12 \\ 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \end{bmatrix}$$

becomes

$$C = (\ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15\ )'$$

The transformation is visualized in 3.1.

3.1 DATA

The face image data includes 80 photo subset of the CMU Face Images Data Set.[14] gray-scale images of 10 people. There are 8 picture of each person, and two photos of each pose: facing right, facing left, looking forward, and looking up.

The fingerprints are from UPEK Fingerprint Database, available at the link in the Bibliography. We use a subset of 10 individuals, scanning one finger 8 times. The number of persons and examples per person match the face dataset.
There are 10 patient classes and two datasets, one with 8 face images per patient, the other with 8 fingerprint scans per patient. All trials choose 10 images from random, one from each category, and categorize those 10 images with the remaining Library. Some experimental tests call for a Library of one patient class, while other methods utilize all of the pictures for the Library. In either case, the test images are never included in the Library during the run. On the next cycle, another 10 images are chosen, and so on until 1000 test images have been classified and averaged. The randomized pattern organizes each patient class the same. There are 8! (40320) combinations from which the 100 runs are sampled. Because the datasets are small, we permuted through the images to create more test/training examples.

3.2 DATA PREPROCESSING

Visual neurons have been theorized as performing a whitening of the initial input such that neuronal responses are highly de-correlated[2]. A whitening step is required to normalize the images before the LCA algorithm converges for these datasets in this sparse modeling application. A whitening transformation converts a set of signals with arbitrary covariance into a set of new signals whose covariance is the identity

Figure 3.2: Raw face data: 10 people, 8 photos, 4 main poses.

Figure 3.3: Raw fingerprint data: 10 examples of a person's fingerprint, 8 scans per finger.

matrix, so each signal has variance 1, and together they are all uncorrelated. It is a decorrelation transformation of the input signal into a white-noise output.[20]. Whitening flattens the frequency spectrum by scaling down the high and low frequencies, and scaling up the mid frequencies. Visually, it greys out the image and brings out the edges. Whitening was applied to all the faces at once, or all the fingerprints of each identity category.

Whitening the images was a necessary preprocessing. The images were whitened in batches, one batch for each finger (8 scans total per finger), so that all the fingerprints from each person were pre-processed as a group. Similarly all of the faces of the same individual were whitened together. With this technique, the same LCA dynamical model accurately identifies images from both data types. No further pre-processing was necessary for L1 pooling over the sparse model to accurately classify the test image 100% consistently after consecutive 1000 test-image trials.

The model worked with Whitening all the fingerprint images together, but required toolbox functions such as binary and thinning to pre-process the fingerprint data. These built-in tools depend on lots of background Matlab code, but are useful. Pseudo-code can be found in the appendix.

Chapter 4

Results

4.1 PRELIMINARY METHODS

Our aim is to show the necessity of non-linear techniques for feature recognition in images. We test straightforward Linear Algebra methods to motivate the need for some new non-linear methods. The percentages are averaged over 1000 trials. It should be noted that 10% is chance.

The first preliminary method to measure the similarity between a test image and the Library images is to measure the dot product of the images as vectors. The dot product, or inner product, is a generalization of the angle between two vectors. We take the inner product of each test image with all of the other images in the Dictionary, for the faces and finger prints.The method measures 10 images at a time, one from each patient class, and those images are set to zero in the Dictionary. We look for the image with the largest dot product, as that implies the smallest angle between the images. The patient identity was compared with the test key identity. For the faces, there were no correct identifications, and only one correct identification of the fingerprints. However, when summing over all dot products by patient class, the total max from the patient classes gives meaningful results. The routine gets 34.4% of fingerprints correct and 28% of faces. A binary threshold and thinning

Classification Routine Results		
	Fingerprints	**Faces**
Inner Product		
single maximum	3.10%	0.30%
sum over class (max)	34.4% †	28%
Euclidean Distance		
single minimum	0.8%	1.8%
sum over class (min)	32.8%	26.6%
Sparse Model LCA		
residual error (min)	52% †	88.8%
L0, sum over class (min)	89.6% †	10.1%
L1 pooling, sum over class (min)	**100%**	**100%**

Figure 4.1: Summary of classification methods, each result is 1000 trials averaged. dagger indicates thinning routine.

routine were applied to fingerprints to improve the results. The routine is indicated in the table with the dagger (†) and matlab code is given in the appendix.

In a similar way, we measure the Euclidean Distance between each vector, and select the minimum distance as a measure of similarity between the vectors. The results over 1000 trials are very poor for choosing the single minimum over the whole set as the method to classify the test image. Summing the results by patient class and selecting the minimum improves the results to 32.8% for fingerprints and 26.6% for faces, see second block of 4.1.

We classified the test data with Sparse Modeling from three different measures. First, because the algorithm works to minimize the residual, we measure the residual error ($(y - Da)$) of an image with the each patient class (7 dictionary images). Choosing the minimum as the classification resulted in 52% correct for fingerprints and 88.8% correct for faces, better than chance of 10%.

The computational routine is constrained to make the representation sparse. The next method we measured was the sparsity of the code by finding the patient class which required the fewest contributions to reconstruct the test image. This is illustrated and described in more detail in Method 2 below. The results over 1000 trials averaged were 89.6% for fingerprints and 10.1% for faces.

Finally, we measured the absolute magnitude of the sparse vector and pooled over the patient classes to classify by L1 Pooling. The results were 100% for faces and 100% for fingerprints consistently over thousands of runs. Therefore, the results of this study found that L1 Pooling was the strongest method for image classification with sparse modeling.

4.2 LCA CONVERGENCE

In the computer model, LCA is a function represented with a dynamical system, integrated with basic Euler Method. The functions u and a work with the input training dictionary D and input test image y to converge to a solution. After a certain amount of loop iterations, the dynamical system becomes extremely likely to converge to the correct solution.

In the first figure, the average score of 10 training images per run are tested over a continuum of loop iterations. This means the algorithm was first run for one loop. Next, it was run for two loops, and the average score was recorded for those and the remaining iterations, as the loop size increased by 1 each time. It can be seen that after approximately 40 steps, the algorithm converges to the correct solution for all 10 images.

To examine the initial path towards successful sparse modeling, we averaged 1000 test-images (100 runs of ten photos each) for each loop size, 1 to 35. The convergence of the algorithm as the step size increases can be seen in the figure. For this example, the fingerprint dataset was used.

4.2.1 Data dependent convergence

The previous figures show the convergence of LCA as a function of itteration size for the fingerprint data. The pattern of convergence is affected by the data type. The figure here shows LCA converging with loop size for the face data. The sparse modeling routine to acheive these plots is described in Method 2: ℓ^1-norm Pooling.

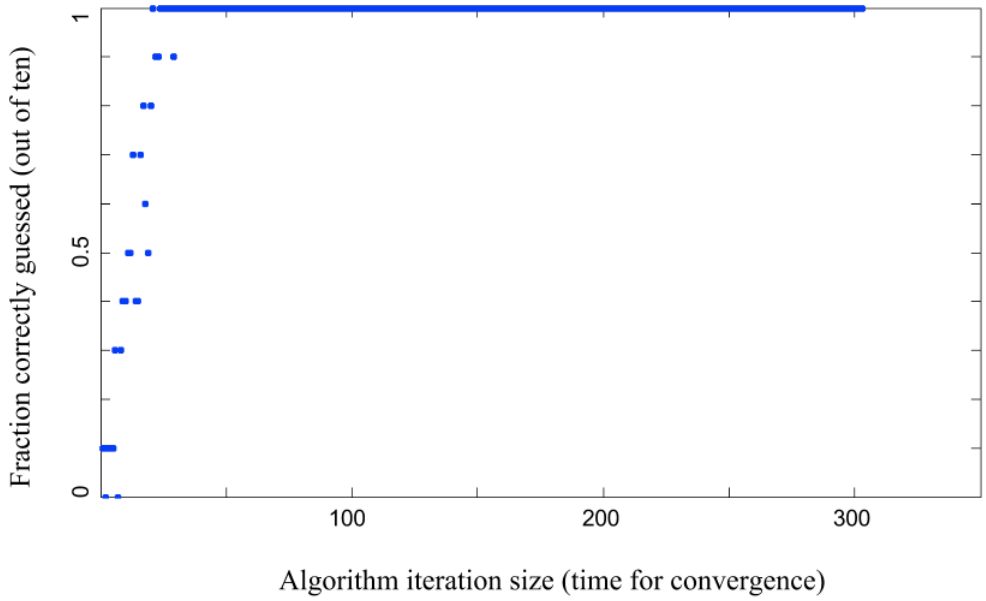

Figure 4.2: LCA convergence stability for loop sizes [1,300], 10 fingerprint test-images averaged per run.

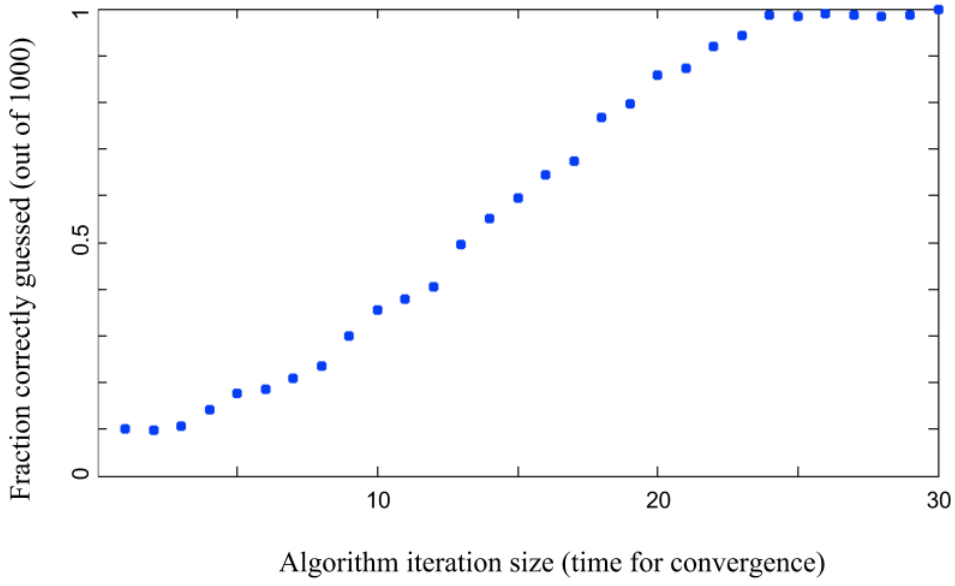

Figure 4.3: LCA convergence for loop sizes [1,30], hard threshhold, 1000 fingerprint test-images averaged per loop size.

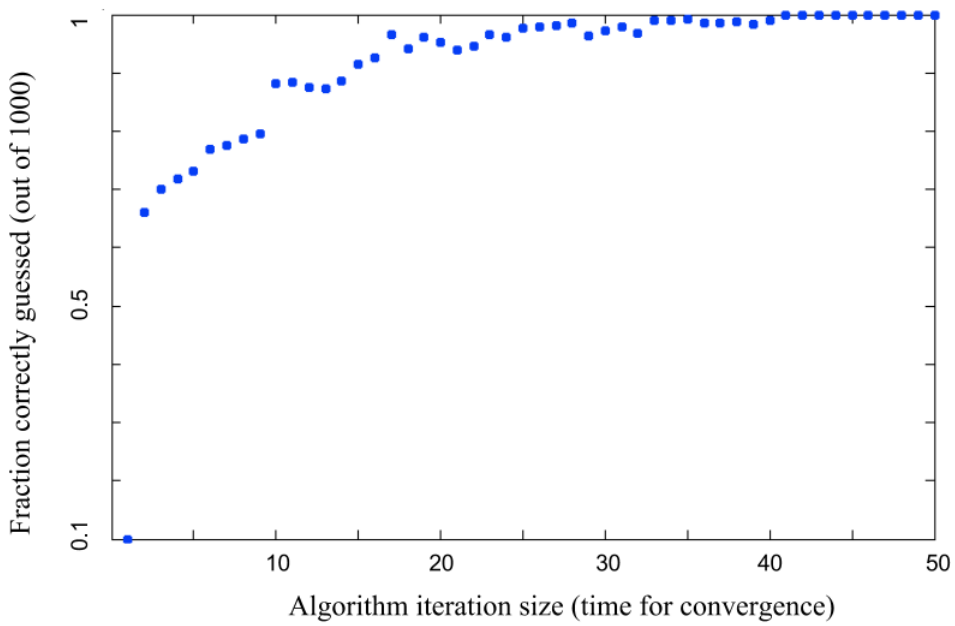

Figure 4.4: LCA convergence for loop sizes [1,30], hard threshhold, 1000 face test-images averaged per loop size

4.2.2 Threshholding

There are a variety of methods for threshholding in the dynamical system. The plots above show the convergence for a type called hard threshholding. The sparse modeling experiments described in the following sections of this report, Method 1 and Method 2, were implemented with hard threshholding. LCA has just two equations in the dynamical system. The first equation is the threshhold. The effect of hard threshholding is to set components with magnitude less than a certain threshhold completely to zero. A hard threshhold is accomplished with the equation

$$a = u. * (abs(u) > \lambda)$$

Take, for illustration, a vector **u**

$$u = [0.3 \quad -0.1 \quad -0.86 \quad 0.04 \quad 1.2 \quad -0.3]$$

With a threshhold value of $\lambda = .2$ applying the hard threshhold gives the result

$$a = [0.3 \quad 0 \quad -0.86 \quad 0 \quad 1.2 \quad -0.3]$$

Now with a threshhold value of $\lambda = .5$, applying the same hard threshhold gives

$$a = [0 \quad 0 \quad -0.86 \quad 0 \quad 1.2 \quad 0]$$

In the experiment, the vectors are much longer and the threshhold value is much smaller, but otherwise this is a good example of how the threshhold induces sparsity in the model.

Another type of threshholding is soft threshholding. Soft threshholding is accomplished with the equation

$$a = (u - sign(u). * (\lambda)). * ((u) > (\lambda))$$

The effect of soft threshholding is to reduce the magnitude of the vector components before applying the hard threshhold. For the example above with vector $\mathbf{u}$ and $\lambda = 0.2$, the soft threshhold returns vector

$$a = [0.1 \quad 0 \quad -0.66 \quad 0 \quad 1.0 \quad -0.1]$$

It can be seen that the soft threshhold technique reduces the absolute value of each entry by the λ, before threshholding according to the value in the original $\mathbf{u}$

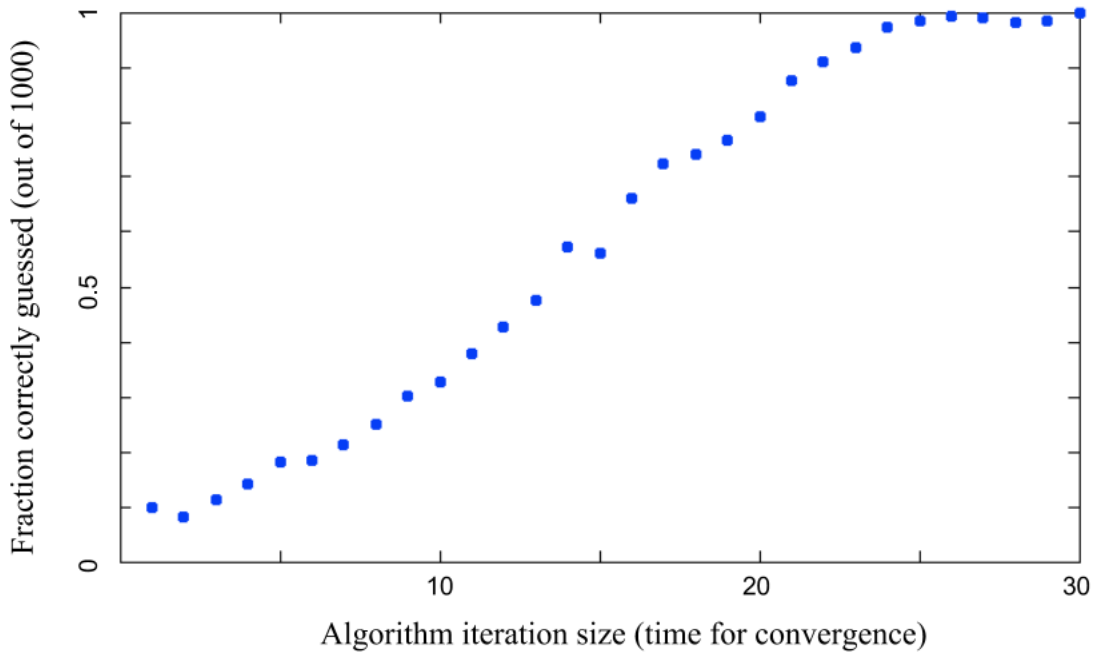

Figure 4.5: LCA convergence for loop sizes [1,30], soft threshhold, 1000 fingerprint test-images averaged per loop size.

In 4.5, the convergence of LCA operated with soft threshholding is shown as a function of iteration size. To demenstrate the utility, soft threshholding was implemented in the section Method 1: Minimum Error.

4.3 METHOD 1: MINIMUM ERROR

This experiment naturally leads to a study of the residual error, since the sparse modeling works to minimize error in order to solve the task. Comparing measurements of the error provides a reliable method to accomplish the sparse modeling task. The residual error is found by computing

$$\varepsilon = \mathbf{x} - \mathbf{Da}$$

where $\mathbf{x}$ is the image and $\mathbf{Da}$ is the reconstruction by sparse modeling. The error ε is a vector with the same dimensions as the image vector. The error is found between the test image and reconstruction with 10 sub-dictionaries, one per patient. The errors are compared by summing up the absolute values of the ε vector. The smallest total sum is declared winner and compared with the test key, which ascends from 1 to 10, as before.

In 4.8, the residual error is applied as the method for image classification. The top left plot shows the total error of the reconstructed test image for each of the ten patient dictionaries. The shortest one has the smallest error, and is awarded the win, as plotted in the bar-graph in the top right. The correct patient is plotted in green dots; the guess in red. Red would appear above or below the line if the guess were incorrect. The bottom left image is the test image and the bottom right is the residual reconstruction.

Interestingly, the fingerprints the correct dictionary consistently has the greatest error among the test images. If we measure the absolulte value between the error and the mean, this value consistently gives the correct answer. The cause is a quandry as to why the minimum residual error (as expected) works to classify images from the

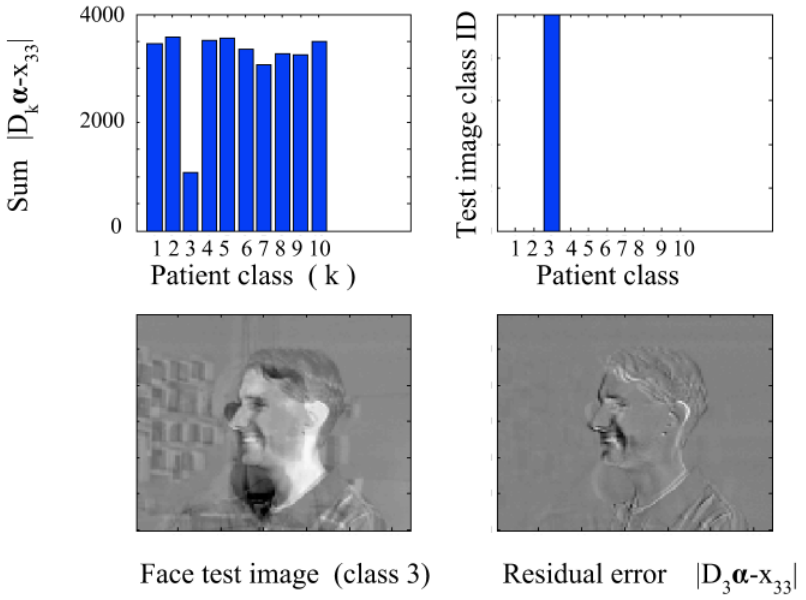

Figure 4.6: Classification by minimum error, test image and residual, 3^{rd} class.

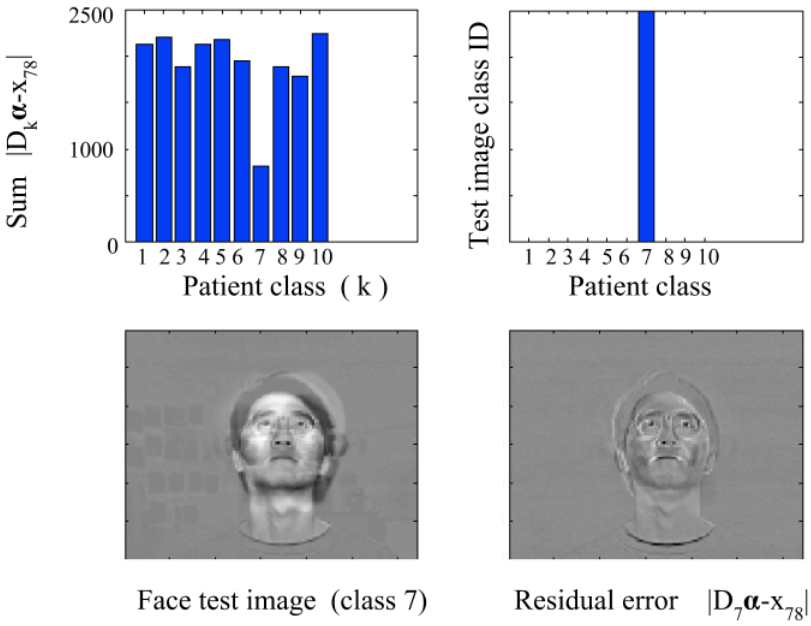

Figure 4.7: Classification by minimum error, test image and residual, 7^{th} class.

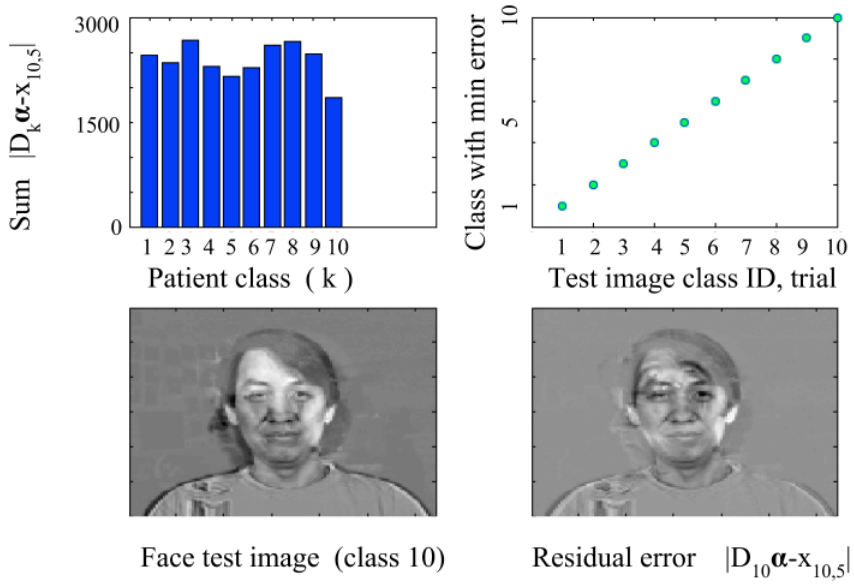

Figure 4.8: Classification by minimum error, test image and residual, 100% after 10^{th} instance in trial.

face dataset, but the maximum residual error (contradictory to expectation) works to classify images from the fingerprint dataset. Perhaps this reason is correlated to the success of the algorithm to achieve the sparsest model in the fingerprints. The LCA algorithm has two terms to optimize: minimize the error, while keeping the representation sparse. With the fingerprints, sparsest representation (described in Method 2) was very successful. The model was able to represent the fingerprint test images with as little as two or three basis vectors from the correct patient dictionary, while it took six or seven images (the max) to represent test images with other patient images. The face dataset, on the otherhand, was unsuccessful when attempting to classify patients with the sparsest model. These findings suggests a tradeoff between a sparse representation (fewest non-zero coefficients in the sparse vector α) and between minimizing the residual error.

4.4 METHOD 2: MINIMIZE L^0-NORM OF α

The sparse modeling problem is solved here with LCA, the dynamical system described above that settles to the solution of the problem

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} ||\mathbf{a}||_1 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\mathbf{a} \tag{4.1}$$

In the first method, we make 10 separate Dictionaries D, each a matrix of seven scans of a single finger. LCA finds a sparse representation of the test image for each subset individual's dictionary, which have 7 images each. In other words, a sparse representation of the test image is found for every instance from each category.

Every trial includes ten fingerprints, one chosen randomly from each patient case. For each fingerprint, a sparse representation is found in terms of the 10 Dictionaries. The representation with the fewest contributions is considered the best representation, and the algorithm's selection.

We test the fingerprints in patient order, so that the correct ID of the first fingerprint is 1, the second fingerprint is person 2, and so on. For example, The fingerprint image is shown in the middle row, left subplot of figure 4.9. As the algorithm guesses correctly, green dots appear along a line of slope one. Mistakes would appear in displaced red dots with the wrong guess on the vertical. These results are recorded in the long subplot at the top of the figure.

At the bottom right, the bar graph shows the number of Dictionary images per patient category used to sparse model the test image. There are ten bars, one for each patient category. The shortest bar represents the category that required the fewest basis contributions, and hence represents the answer of the algorithm. In the

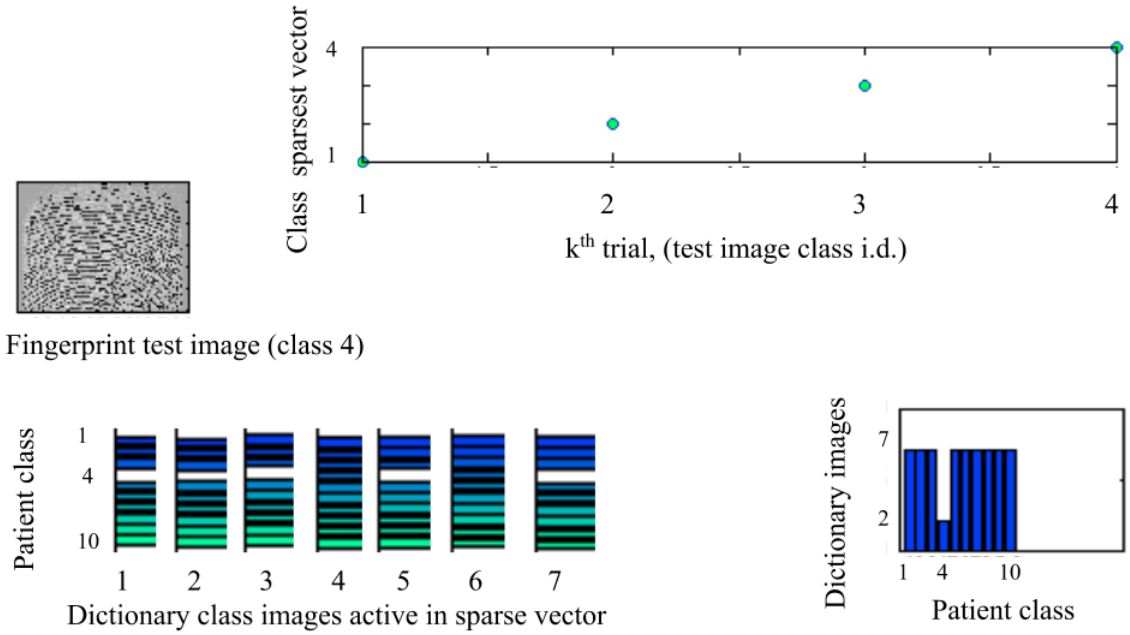

Figure 4.9: Sparse Modeling with L^0 norm and separate dictionary for each patient class, 4^{th} patient class test image. top: correct or incorrect guess; middle: test image. bottom left: activity of patient dictionaries; bottom right: count of D_i needed for x_i reconstruction for each patient class.

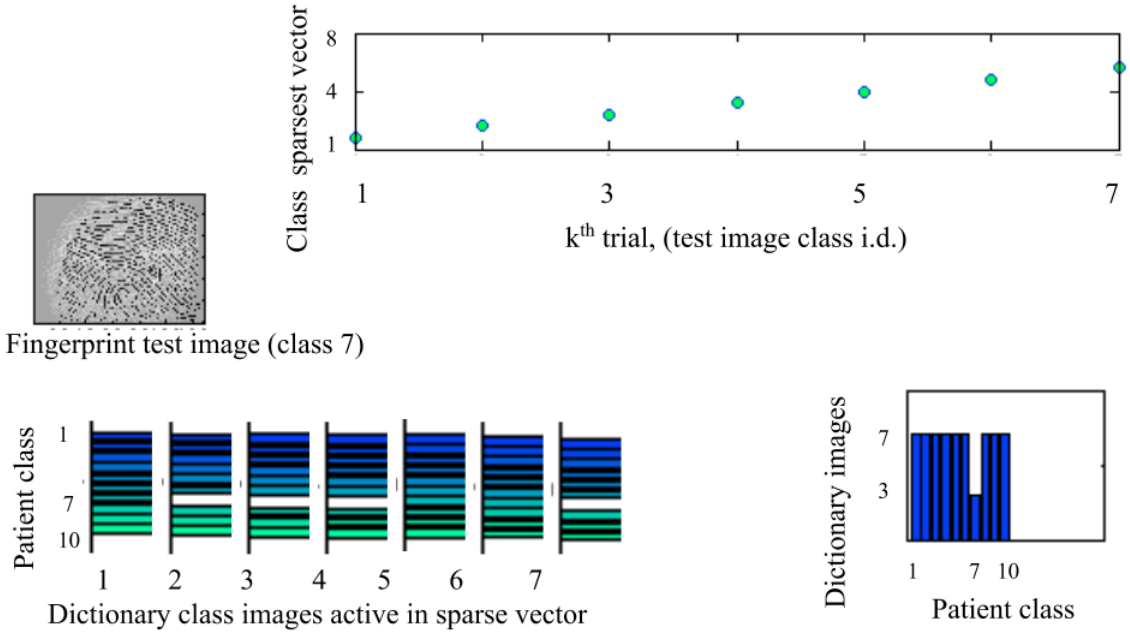

Figure 4.10: Sparse Modeling with L^0 norm; 7^{th} fingerprint example in trial.

top subplot, for a given test image this corresponds to a round red dot plotted on the vertical axis, that is generally concealed by the green dot (correct answer key).

This sparse representation is accounted for in the longer subplot on the bottom right. White gaps can be seen in the 4^{th} patient class, indicating a sparse representation of the fingerprint in the 4^{th} patient's unique basis. The sparse model represented the test image with two dictionary images from patient class 4, but with all seven images from the remaining patient classes. This can be seen in the bar graph on the bottom left, where the bar corresponding to patient 4 has only two contributions. When the algorithm ran this 4th fingerprint against the other dictionaries, all seven images were needed, and the representation was clearly most sparse for the 4th person's dictionary.

In 4.10, the 7^{th} patient's fingerprint is represented most sparsely with the correct identity's fingerprint basis.

4.5 METHOD 3: ℓ^1-NORM POOLING

For a given trial, ten pictures were extracted at a time and represented as a sparse linear combination of the remaining 70 images, solved with the same LCA function as above. Now, all 70 images are driven to compete against eachother for representation. This drives up the number of contributions from similar images, while working to make the represntation in the entire basis sparse. To count the contributions of each test image, the corresponding magnitudes of the $||x||_1$ norm are added. This L^1 pooling method counts the largest total sum over the 7 images, choosing that patient category as the algorithm's guess. This data is visualized in the bar graph at the bottom left.

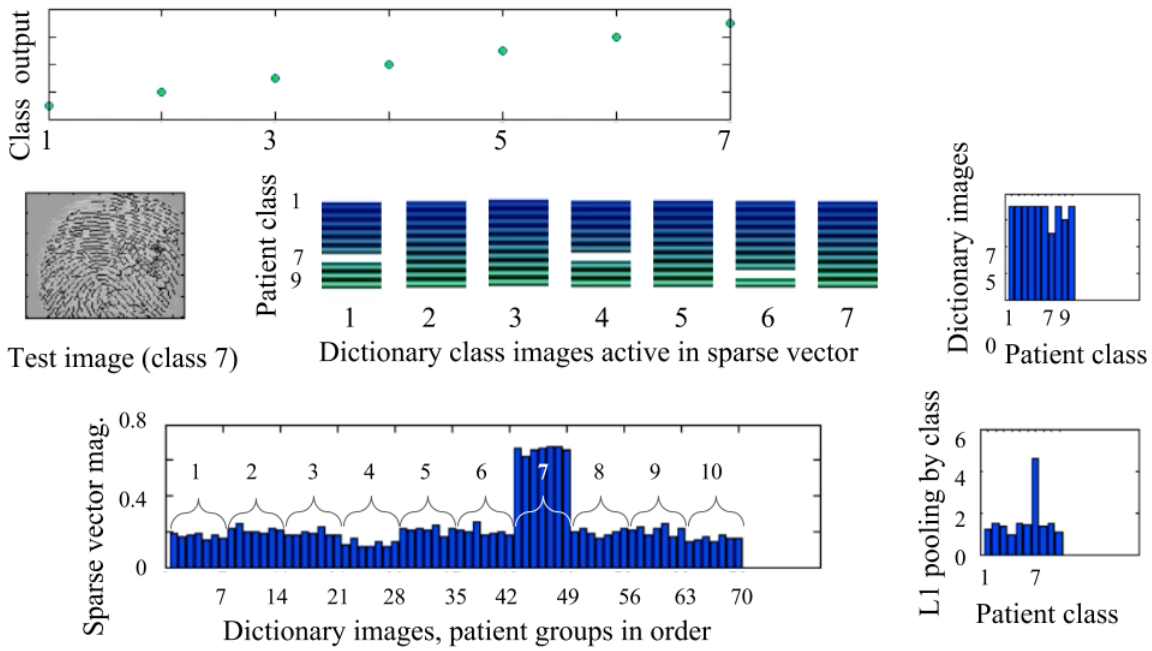

Figure 4.11: Sparse Modeling with both L^0 norm and L^1 pooling; 7 correct tests images. top: correct classification with two routines. middle: classification by minimum L0 norm of sparse vector, with patient class dictionary. bottom left: magnitude of sparse vector elements. The dictionary includes all ten patients. first seven bars are of patient 1, etc. bottom right: L1 pooling of sparse vector by patient class, win goes to the max.

The answer is also plotted along the same dotted path and compared with the key (of slope 1) in the top subplot. Mistakes would appear as dots above or below the line.

The exact same function of LCA can be run with both models simultaneously, with the same model parameters. In this case, the threshold λ, integration stepsize, and model parameters are kept constant. It can be seen that the same function of LCA solves both methods, giving accurate sparse representations of the data. This evidence

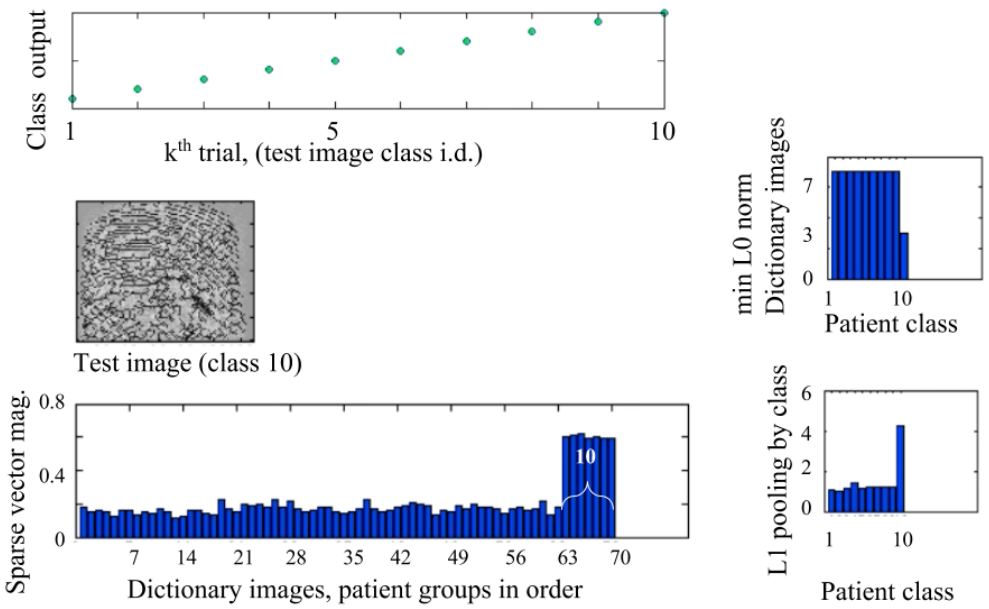

Figure 4.12: Sparse Modeling with both L^0 norm (middle) and L^1 pooling (bottom); 10 correct tests after one complete trial.

should emphasize that sparse modeling is a robust technique for modeling data.

4.5.1 Parameter-Based Performance

Full statistical exploration of the model would require extensive work on larger datasets, and large number of trials run ergodically through the phase-space of the hyperparameters. With limited data and parameter exploration, the optimal parameter settings of Method 2 (L0-sparsity) for correct fingerprint identification were found with $\lambda = 0.073,\ t = .01,\ h = .0000001$, and 100 LCA iterations. The threshold λ imposes sparsity on the model. The ratio h/t determines the step-size in the Euler integration of LCA. The iterations are the number of steps before the sparse vector is measured. With these settings, the $L0$ model had 96.95% accuracy over 1000 images (100 runs, 10 photos per trial).4.13 Method 3, L1 pooling, had 99.47% accuracy with the same settings. Keeping the step-size the same, and setting $\lambda = 0.003$ (as well as $\lambda = 0.001$),

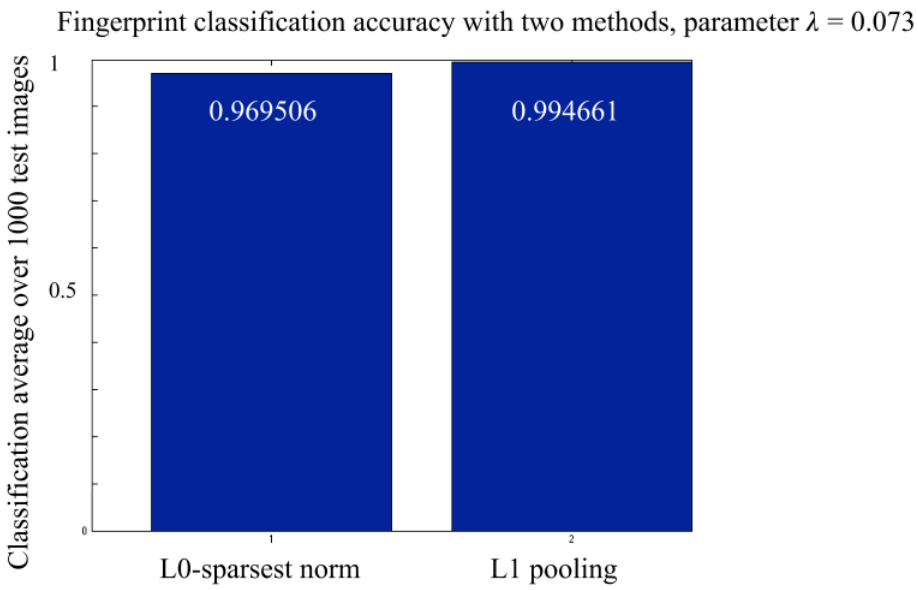

Figure 4.13: L^0 norm and L^1 pooling; results after 1000 fingerprint test images, with $\lambda = 0.073$, $\frac{h}{t} = .00001$.

L1 pooling consistently had 100% accuracy.

4.6 FACES, FINGERPRINTS, AND MULTI-MODAL APPLICATION

We apply Method 3, L1 pooling to correctly identify faces. No special subroutines are needed, only the whitening pre-processing. Each patient has four poses; chin left, right, up, and forward. There are two pictures of each pose, with noticeable differences in facial expression and precise spacial location.

One image from each patient class is taken from the dataset and is reconstructed with the dictionary and sparse model. For any given test image, seven of the 70 dictionary images match the identity. Nineteen of the pictures are of the same pose. Eighteen

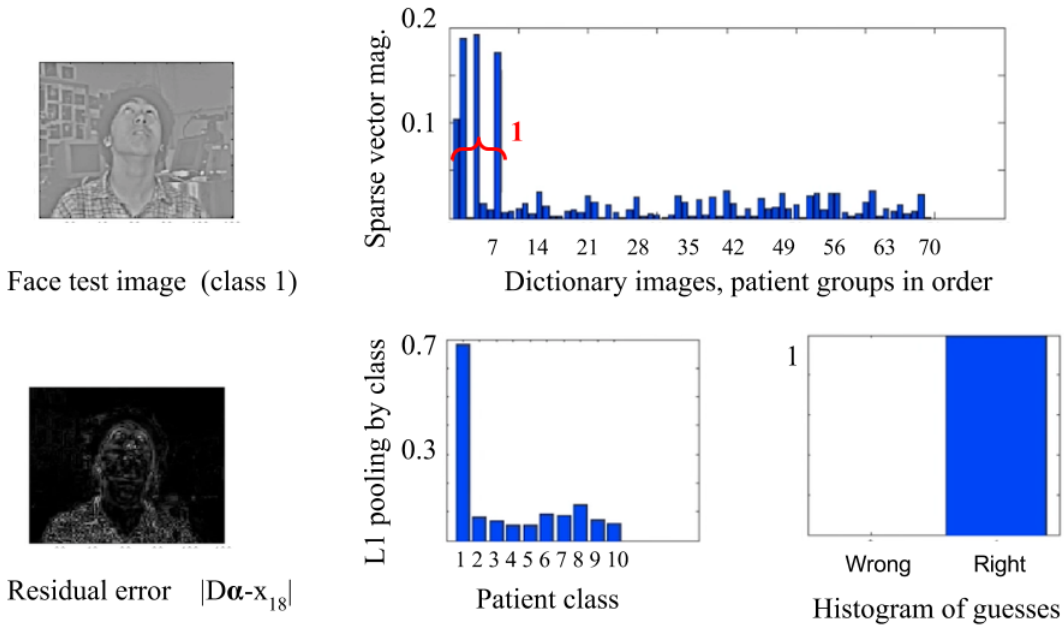

Figure 4.14: ℓ_1 pooling, faces, 1^{st} instance in trial.

of those with the same pose have the wrong patient identity. One image is a match to the test image, in identity and pose. LCA solves the sparse optimization problem. Our network undergoes ℓ_1 pooling over α (summing the absolute values) for each patient, resulting in a new vector that describes the contribution of each patient class to the reconstruction.

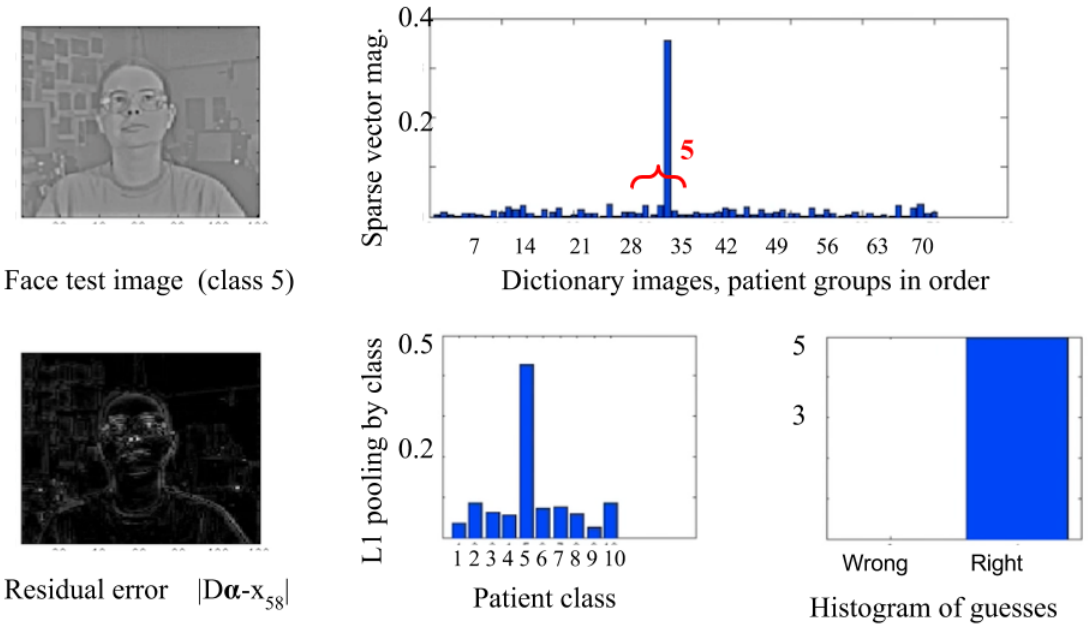

Figure 4.15: ℓ_1 pooling, faces, 5^{th} instance in trial

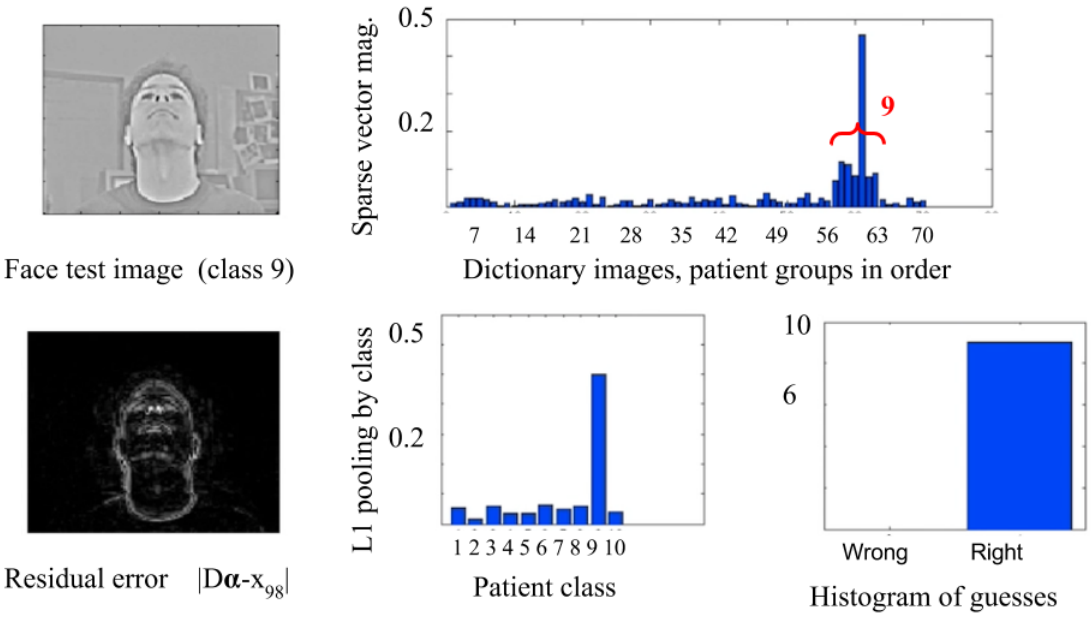

Figure 4.16: ℓ_1 pooling, faces, 100% after 9^{th} instance in trial

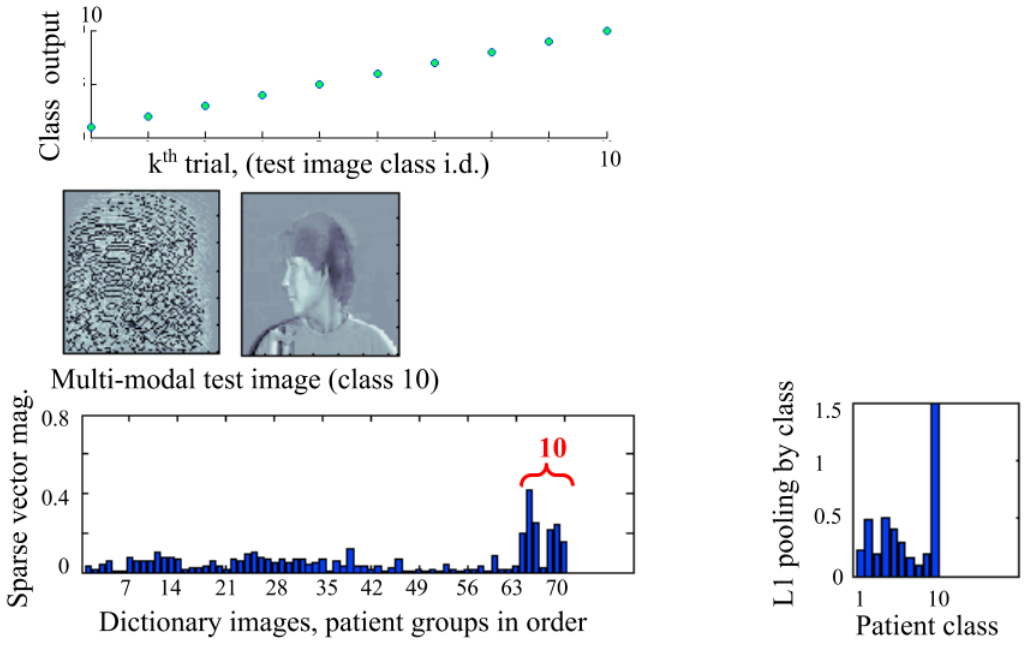

Figure 4.17: ℓ_1 pooling, fingerprint-face vector, 100% after 10^{th} instance in trial

4.6.1 Sensor Fusion

Fingerprint and face data were acquired with completely different camera systems. We preprocess the fingerprint and face data separately from each other. The LCA sparse modeling algorithm can handle an image vector that stacks the data. The idea of fusing the data is relevant because it allows machine learning algorithms to potentially learn more by seeing more at once. We stack a fingerprint vector and face vector into one multi-sensor, multi-modal image, and find that the LCA model correctly identifies the new vector.

The model is reliable with the addition of large amounts of random noise. The figure has an example of multi-modal data with 60% random noise added to the signal. The model consistently identifies correctly all of the test images.

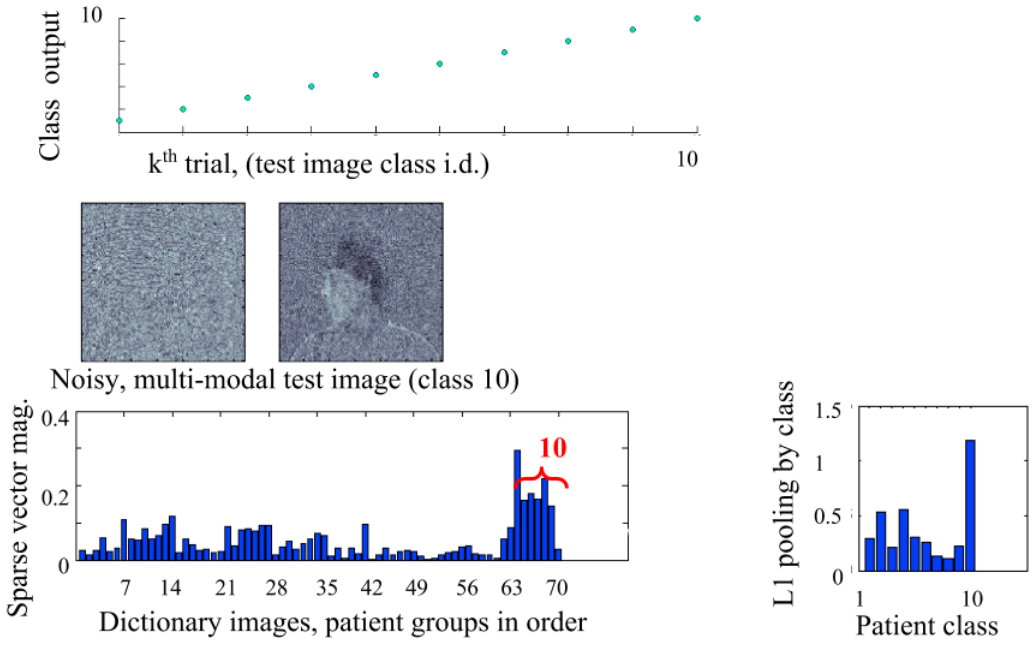

Figure 4.18: ℓ_1 pooling, fingerprint-face vector with 60% random noise, 100% after 10^{th} instance in trial

4.7 CONCLUSION

According to MIT Professor Thomas Malone, the important question regarding computers and medicine is: "How can people and computers be connected so that - collectively- they act more intelligently than any person, group, or computer has ever done before?[16]

In this work, we have described how a sparse modeling framework can accomplish a number of important tasks needed for medical imaging physics. We demonstrated how to create a sparse dictionary using digital images, to classify unknown image data using sparse approximation. We showed that digitical images can be vectorized and used as columns in a dictionary matrix, then used this dictionary to sparsely decompose an unknown input image. By pooling over the dictionary atoms used in the sparse approximation, we can classify the input image. Photographs of faces at various poses and digitally scanned fingerprints were used to demonstrate the versatility and robustness of the technique, even to noise. Typically faces and fingerprints both require hand-tuned subroutines to extract features, relying on multiple layters of feature extraction. in our technique, no feature extraction is required.

The promise of this technique is to extend to other domains where traditional feature extraction techniques are not obvious or have failed historically such as high-dimensionaly and multi-modal medical image data. Ultimately, treatment systems should integrate patient diagnostic treatment and safety all into a single mathematical framework to allow advanced machine learning techniques to aid in future health challenges. Sparse moedling's ability to remain agnostic about the particular image or data domain allows for great utility in exploring new applications of multi-modal

medical datasets.

Chapter 5

Appendix

5.1 PSEUDOCODE

The following are examples of psuedo-code in Matlab:

Whitening

```
fX = fft(fft(L,[],2),[],3);
spectr = sqrt(mean(abs(fX).^2));
L = ifft(ifft(bsxfun(@times,fX,1./spectr),[],2),[],3); L
```

Inner Product

```
for i = 1:n

   a=dot(y,L(:,i)];

end
```

LCA and L1 Pooling:

```
a = LCA(y, D, lambda);


for i=1:10
    b = [b sum(abs(a(find(key==i))))];
 end


[b1,b2]=max(b)
t=[t b2==test_key(k)]
bar(b)
bar(abs(a))
hist(t)



function [a, u] = LCA(y, D, lambda)
t=.01;
h=.000001;
d = h/t;
u = zeros(size(D,2),1);

for i=1:100
    a=u.*(abs(u) > lambda);
    u =   u + d * ( D' * ( y - D*a ) - u - a  ) ;
end
```

```
alpha = LCA(y, D, lambda);

for i=1:(# of patients)
     Take absolute value of alpha, sum over photos by patient class
 end
 Find the ID who contributes the maximum to alpha
 Compare the result to the ID of patient test image (y)

function [a, u] = LCA(y, D, lambda)
t=.01;
h=.000001;
d = h/t;
u = zeros(# of photos);

for i=1:100
    alpha=u.*(abs(u) > lambda);
    u =   u + d * ( D' * ( y - D*alpha ) - u - alpha  ) ;
end
```

Min L0 Norm (Cardinality of Support):

```
     b=sum(abs(a)>0)

    [b1,b2]=min(b);
```

```
t=[t  b2==patient_names_key(k)];


rate = sum(t)/length(t)
```

Thinning sub-routine:

b1=im2double(imread(a1)); b1=imresize(b1,[120,128]);

b1=b1(:,:,1)¿0.3;

b1=bwmorph(b1,'thin','inf');

b1= b1;

BIBLIOGRAPHY

[1] Radiation therapy staffing and workplace survey 2014.

[2] Joseph J Atick and A Norman Redlich. Towards a theory of early visual processing. *Neural Computation*, 2(3):308–320, 1990.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] Chi-hau Chen. *Computer vision in medical imaging*, volume 2. World scientific, 2014.

[5] D Daniel. Introduction to sparse representation, 2016.

[6] Mark A Davenport, Marco F Duarte, Yonina C Eldar, and Gitta Kutyniok. Introduction to compressed sensing. *Preprint*, 93(1):2, 2011.

[7] William Edward Hahn, Stephanie Lewkowitz, Daniel C Lacombe Jr, and Elan Barenholtz. Deep learning human actions from video via sparse filtering and locally competitive algorithms. *Multimedia Tools and Applications*, pages 1–14, 2015.

[8] Richard R Hamming. *Art of doing science and engineering: Learning to learn*. CRC Press, 2003.

[9] Environmental Health and Safety. Misadministration, 2008.

[10] Jeremy Howard. Enlitic: Data driven medicine, 2015.

[11] Judson P Jones and Larry A Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of neurophysiology*, 58(6):1233–1258, 1987.

[12] Janaki Krishnamoorthy, Adela Salame-Alfie, and John OConnell. An analysis of radiation therapy medical events in new york state: the role of the state radiation programs in patient safety. *Health physics*, 106(5):S71–S77, 2014.

[13] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10):95–103, 2011.

[14] M. Lichman. UCI machine learning repository, 2013.

[15] Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*, 2014.

[16] Thomas Malone. Social cognition and collective intelligence, mit 150 symposium: Center for brains, minds and machines, 2011.

[17] Anke Meyer-Baese and Volker J Schmid. *Pattern recognition and signal analysis in medical imaging*. Elsevier, 2014.

[18] MPCRlab. Face recognition with live camera, 2015.

[19] Andrew Ng. Image classification with sparse coding, 2010.

[20] Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487, 2004.

[21] Irina Rish and Genady Grabarnik. *Sparse Modeling: Theory, Algorithms, and Applications*. CRC Press, 2014.

[22] Christopher Rozell, Don Johnson, Richard Baraniuk, and Bruno Olshausen. Locally competitive algorithms for sparse approximation. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 4, pages IV–169. IEEE, 2007.

[23] Christopher J Rozell, Don H Johnson, Richard G Baraniuk, and Bruno A Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563, 2008.

[24] C.J. Rozell, D.H. Johnson, R.G. Baraniuk, B.A. Olshausen, and R.L. Ortman. Analog system for computing sparse codes, 2010. US Patent 7,783,459.

[25] Peter F Schultz, Dylan M Paiton, Wei Lu, and Garrett T Kenyon. Replicating kernels with a short stride allows sparse reconstructions with fewer independent kernels. *arXiv preprint arXiv:1406.4205*, 2014.

[26] John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas S Huang, and Shuicheng Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[27] Shaoting Zhang, Yiqiang Zhan, and Dimitris N Metaxas. Deformable segmentation via sparse representation and dictionary learning. *Medical Image Analysis*, 16(7):1385–1396, 2012.