ABSTRACT

Title of dissertation: LEARNING FROM MULTIPLE
VIEWS OF DATA
Abhishek Sharma, Doctor of Philosophy, 2015

Proposal directed by: Professor David W. Jacobs
Department of Computer Science

This dissertation takes inspiration from the abilities of our brain to extract information and learn from multiple sources of data and try to mimic this ability for some practical problems. It explores the hypothesis that the human brain can extract and store information from raw data in a form, termed a common representation, suitable for cross-modal content matching. A human-level performance for the aforementioned task requires - a) the ability to extract sufficient information from raw data and b) algorithms to obtain a task-specific common representation from multiple sources of extracted information. This dissertation addresses the aforementioned requirements and develops novel content extraction and cross-modal content matching architectures.

The first part of the dissertation proposes a learning-based visual information extraction approach: Recursive Context Propagation Network or RCPN, for semantic segmentation of images. It is a deep neural network that utilizes the contextual information from the entire image for semantic segmentation, through bottom-up followed by top-down context propagation. This improves the feature representation of every super-pixel in an image for better classification into semantic categories. RCPN is analyzed to discover that the presence of bypass-error paths in RCPN can hinder effective context propagation. It is shown that bypass-errors can be tackled by inclusion of classification loss of internal nodes as well. Secondly, a novel tree-MRF structure is developed using the parse trees to model the hierarchical dependency present in the output.

The second part of this dissertation develops algorithms to obtain and match the common representations across different modalities. A novel Partial Least Square (PLS)

based framework is proposed to learn a common subspace from multiple modalities of data. It is used for multi-modal face biometric problems such as pose-invariant face recognition and sketch-face recognition. The issue of sensitivity to the noise in pose variation is analyzed and a two-stage discriminative model is developed to tackle it. A generalized framework is proposed to extend various popular feature extraction techniques that can be solved as a generalized eigenvalue problem to their multi-modal counterpart. It is termed Generalized Multiview Analysis or GMA, and used for pose-and-lighting invariant face recognition and text-image retrieval.

LEARNING FROM MULTIPLE VIEWS OF DATA

by

Abhishek Sharma

PhD dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Defense Committee:
Prof. David W. Jacobs, Chair/Advisor
Prof. Larry S. Davis
Prof. Yiannis Aloimonos
Dr. Oncel Tuzel
Prof. Min Wu, Dean's Representative

To Anamika Dubey, for her being the reason to take up research as a career.

Acknowledgments

First and foremost my deepest gratitudes go to my advisor Professor David William Jacobs, my academic father. His contribution and support for the last five years, of which graduate life was only a part, and many years to come in my life cannot possibly be expressed in the limited framework of language. However staying true to the spirit of my dissertation, I will still dare to obtain a lossy-projection of my gratitudes towards David into the language space. Following is an $\ell 1$-regularized N-gram projection of David's contribution - awesomely brilliant, unassumingly selfless, humiliatingly humble, scrupulously ethical, inspirationally curious, perennially available, ... David's contributions are far deeper than this dissertation for the values that I learned from him will stay with me for life.

I would like to extend my deepest regards to Prof. Larry S. Davis, Dr. Oncel Tuzel, Prof. Ming Wu and Prof. Yiannis Aloimonos for serving as my committee and providing me with valuable comments and feedback about this dissertation. I am thankful to Prof. Hal Daumé III for serving as a committee member for my proposal and various technical discussions about out-of-the-box ideas to change the world of text and image based learning.

I would like to thank Dr. Oncel Tuzel for providing me with the opportunity to work with him at MERL on the semantic segmentation project. His unconditionally kind support, razor-sharp troubleshooting skills and steadfastness towards the idea of using neural networks for semantic segmentation was very inspirational and helpful. His unexpected hands-on support with coding was a bliss when I was struggling to balance my duties as an intern and an active political volunteer. The lunch-time eating-out sessions with Oncel were a pleasure to the palate as well as the brain for they involved discussions on topics ranging from research to revolutions.

I find myself extremely lucky to have worked with Dr. Trishul Chilimbi, Yutaka Suzue and Dr. John C. Platt at Microsoft Research Redmond as a summer inter on large scale object recognition. This internship was my first hands-on experience with deep neural networks and large-scale parallel computing software development and Trishul and John helped me overcome the black-magic myth of training deep neural networks. Yutaka was extremely patient and supportive while correcting my mistakes and teaching me the usefulness of assembly level programming and the hidden treasures of Visual Studio, these skills will go a long way with my career. I would like to thank Prof. Sanja Fidler for a fruitful summer of research with her at Toyota Technological Institute Chicago. I truly admire her research insight, kind-nature and her remarkable capability of putting together awesome GUIs for any task within hours.

A graduate life is incomplete without fellow travelers and friends. I find myself blessed with some very talented graduate students and friends along the way. The Abhishek and Sumita duo lovingly extended the opportunity of technical discussions with Abhishek, most of them were a great learning experience for me, and Sumita's awesome food with their sheer passion and responsibility towards social reforms in India. The fortnightly dinners at their home were a cogent, inspirational and palatable respite from the otherwise routine graduate life. I would like to acknowledge Jonghyun Choi, Murad-Al Haj and Behjat Siddiqui for successful collaborations and learning experiences. I thank Angjoo Kanazawa for numerous technical and general discussions we had together. I truly admire her passion for research, steadfastness, industry and positive attitude towards life. I would like to thank Bharat Singh, Ejaz Ahmed and Udayan Khurana for being there with me in the times of emotional distress and several wonderful dining experiences coupled with cogent discussions on various topics.

I would like to thank all the UMIACS and CS staff members for being so supportive and helpful for the last five years, without their timely urgent support I would have surely missed a couple deadlines for submission. A special thanks to Jennifer Story and Fatima Bangura for saving me from numerous administrative goof-ups that I kept making throughout my graduate life, without them I really cannot think of graduating.

I owe my deepest thanks to my parents and my brother who have always stood by me unconditionally and lovingly. They bore the pain of living separately from me, yet always encouraged me to take as much time as required to complete my studies in flying colors. I was always challenged by my father to do better and for that I owe him my entire career and success, academic and otherwise.

Table of Contents

List of Figures

List of Tables

List of Notations

$\mathbf{x}$	vectors are boldface, lowercase letters
A	Matrices are italic, uppercase letters
$\mathbf{v}_m$	subscripts are used to index modalities for vectors, matrices and scalars
$\mathbf{v}^i$	superscripts are used to index instances
x	scalar variables are italic, lowercase letters

Chapter 1

Introduction

The ultimate goal of Machine Learning and Artificial Intelligence is to mimic human reasoning and learning capabilities. One of the most remarkable feats of the human brain is its ability to extract and combine relevant information from multiple sources of data. The machine learning literature speaks of this indispensable ability as extracting *content* (information) from multiple *modalities* (sources). For example, we can readily combine written instructions and 2D diagrams to assemble an office desk. To date, there is no machine that can perform this task for even moderately complex structures. This simple task illustrates the following challenges that arise due to multi-modality:

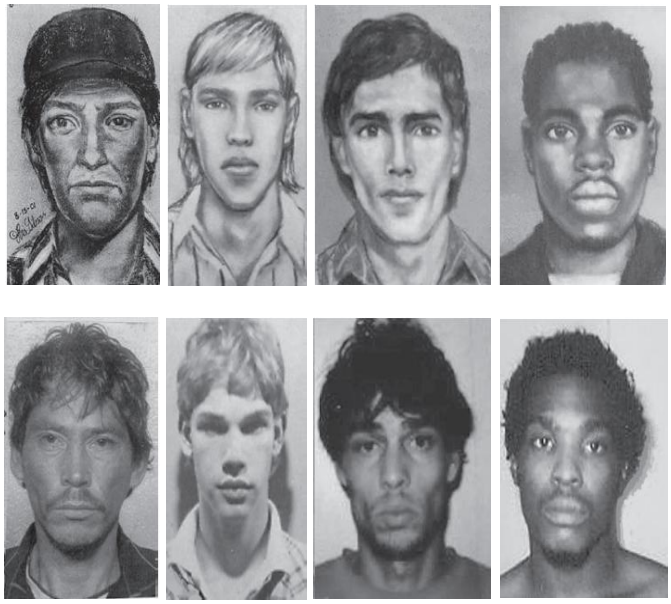
1. **Content extraction** - This refers to the process of extracting task-dependent useful information from data. In the desk installation example, it refers to the ability to understand the illustrative diagrams and the written textual instructions from the manual.

2. **Cross-modal content matching** - This refers to the process of possible transformation of the extracted contents followed by matching or relating across different modalities. In the desk installation example, it refers to relating one visual modality, a sketch of a wrench, to the 3D wrench we look at, as we assemble the desk.

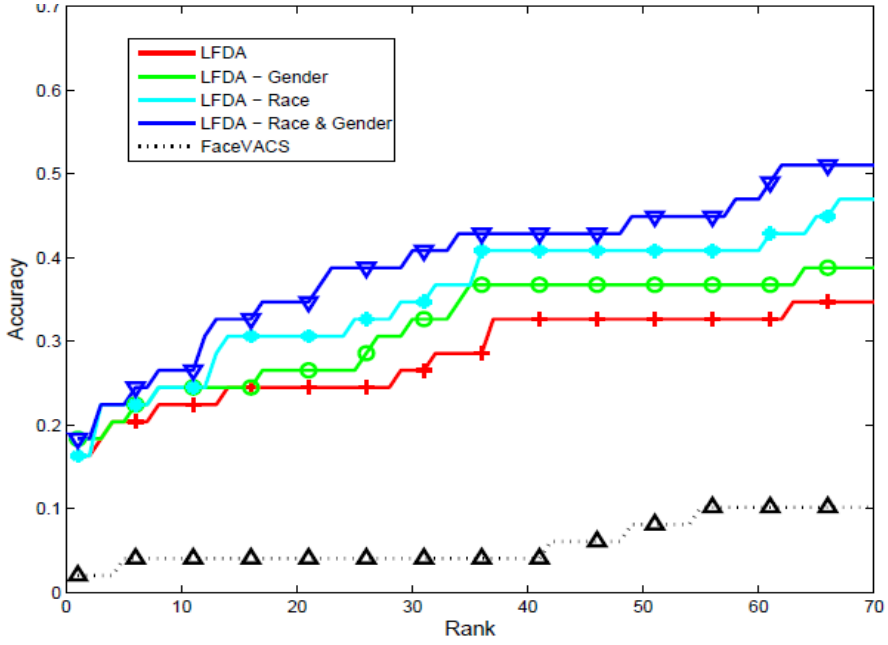
This work attempts to address the aforementioned problems by developing models for visual content extraction from images and algorithms to obtain task-dependent content from multi-modal features.

The cross-modal content matching problem involved in the desk-installation example is a case where the required human effort and time is manageable but there are a number of situations where it is impossible for humans to carry out the task due to the amount of required effort and time. For example, sometimes it is necessary to match a forensic sketch, based on verbal description, to a very large database of face images. Fig 1.1 shows that we can easily figure out the face image corresponding to the given forensic sketch despite a great difference in the overall appearance. In this case, content is the appearance of the person and different modalities are CCD image and forensic sketch. But, a machine-based automatic algorithm performs very poorly on the same task, Fig. 1.1b. As a second example, in Fig 1.2, we can easily tell that the first image is the closest match to the semantics conveyed by the sentence among the five images in terms of the spatial layout of the scene. In this example, despite completely different ways (image and text) of conveying the information, we are able to relate them easily. The state-of-the-art systems of sentence generation from an image and text-based image retrieval are far worse than human performance on real-life images [54].

All these different instantiations of cross-modal content matching are so easy that anyone can perform them with great accuracy and ease, but performing them on a database with thousands of samples becomes totally impractical, if not impossible. Therefore, we need machines to perform these tasks for us. The two key requirements for a machine



(a) Sketch-face pairs



(b) Computer performance

Figure 1.1: (a) Forensic sketch and image pairs of some suspects drawn by sketch artist Lois Gibson. Each column contains a pair of sketch (first row) and the corresponding photo (second row) of the same subject. The sketches are drawn based on the verbal description given by the victim. (b) The result of automatic sketch-face matching for 49 subjects, taken from [56]. **LFDA** is their approach, **LFDA-Gender** is the approach with a gender filter, **LFDA-Race** is the approach with a race filter and **LFDA-Gender-Race** is the approach with a gender and race filter. faceVACS is a commercial face recognition software. It can be see that the automatic algorithms are only around 50% accurate for rank-70 matching, which is far from acceptable. Whereas, humans are far better at this task for limited size datasets.

Figure 1.2: Five images containing the same object ie aeroplanes, the difference is in terms of the spatial layout of these objects in the scene. The task is to find the closest matching image for the descriptive sentence. The images and sentence come from UIUC sentence dataset [94].

for satisfactory performance are speed and accuracy. From the aforementioned examples one can observe that there is a huge difference between human and machine performance. Although these problems look entirely different, they all are simply different instantiations of content extraction and cross-modal content matching. Here, we stress upon the fact that sufficiently rich task-dependent content extraction from each modality is also crucial for the final performance of the subsequent cross-modal content matching. For example, [56] showed that local-gradient based discriminative feature extraction from the sketches and faces is crucial for sketch-face matching and it improves the performance from 4% to 20% rank-1 accuracy when compared against traditional features. Similarly, [54] showed that the use of deep Convolution Neural Network based features from images and deep Recurrent Neural Network based text-generation led to twice better accuracy for image retrieval and sentence generation as compared to traditional features.

We have already seen that humans are very good at content extraction and cross-modal content matching. A natural question is: Why? To date, there is no concrete theory that offers a complete explanation, yet there are some popular hypotheses that offer possible explanations. One such hypothesis is that our brain extracts and stores content from multi-modal data in a canonical form [24]. Thus, it facilitates seamless cross-modal content matching. Informally, this can be termed the *common representation hypothesis*. It is useful and interesting in cases with more than two modalities because it provides a common framework for representing and working with multiple modalities.

1.1 From ideas to numbers

So far, we spoke of concepts such as: task, content and modalities in an abstract sense in order to facilitate intuition. In this section, we model these concepts as mathematical objects with well defined operations to facilitate analysis on a machine.

1. **Task** - A task, just as with its normal definition refers to a set of operations with a desired goal. For example, given a face image and a forensic sketch, the task is to tell whether they come from the same person or not. Mathematically, it is a function that takes in arguments and outputs the result.

2. **Content** - Content refers to the task-specific representation of information required to complete the task. Most machine learning algorithms operate on vectors spaces of real numbers. Therefore, it is natural to represent the content as a vector with each dimension describing some part (attribute) of the content. Such a vector and the associated vector space are formally known as feature vector and feature space, respectively. For example, a gray-scale face image can be represented as a vector of pixel intensities. Similarly, a document can be represented as a bag-of-word vector with each dimension being the count of occurrence of a particular word in a dictionary. Note that the usefulness of the content depends on the task. For example, gray-scale face images are sufficient for face recognition under controlled lighting condition, but it is required to obtain gradient based features from gray-scale images for satisfactory performance under varying illumination. Therefore, we can see that useful content extraction within a modality is also important and crucial for the success of cross-modal content matching.

3. **Modality** - We can have several different representations for the (approximately) same content, depending on our requirements and input data. In some cases, it is

trivial to find a mapping to bring different representations to a common representation. For example, coordinates of points in the 2D plane can either be expressed in Cartesian or Polar coordinate system and the mapping between these two is known to us. However, in some situations the mapping between different representations is not straightforward to obtain. Under this situation, different representations of the same content constitute modalities. For example, the task of face recognition can either use pixel intensities or visual attributes (gender, race, hair color, skin color, etc.) as feature vectors. These two modalities are completely different and span different feature spaces, yet both contain information that can be useful for face recognition. Therefore, these two representations constitute two different modalities, for the task of face recognition.

With these definitions in hand, let's try to understand the requirement of content extraction and a common representation. To facilitate intuition and provide a visual aid we will take up the examples of text-based image retrieval and pose-invariant face recognition. In text-based image retrieval, the task is to retrieve an image from a database using a verbal query. We want the retrieved image to be a close match to the concept described in the query. Although, a vector of all the pixel values of the image constitutes a representation scheme, but it is not a good representation of the task-dependent visual content such as objects, scene type and geometric layout. Similarly, a string of characters is also a representation scheme for textual query, but it is not a useful representation for the aforementioned task. A pair of representations for image and text that contains the required content could be a SIFT histogram and a Bag-Of-Word feature, respectively. Therefore, both of them can be represented as vectors. However, it is possible to have different dimensions for the two feature vectors. In this situation, it is not even possible to calculate a distance between the image and text feature vectors unless we have a learned metric. Suppose we decide to have the same dimensions for the image and text feature vectors in order to facilitate simple Euclidean distance based similarity. Unfortunately, the Euclidean distance will not give any meaningful information, though the feature vectors have similar content, they are entirely different in terms of presentation of the content. Therefore, a common representation is required to relate samples across modalities.

Similarly, we can regard a face image as a vector in $\Re^{\mathrm{D}}$. The coordinate axes defined for each pixel will constitute a *representation scheme* ($\mathcal{S}$) for the face which is basically the set of column vectors of an identity matrix in $\Re^{\mathrm{D}}$ space. Corresponding pixels across different subjects' faces roughly correspond to the same facial region in the absence of pose difference and controlled lighting conditions. This *feature correspondence* facilitates comparison. In fact, feature correspondence is essential for comparison based on any learned model. For faces especially, it has been shown to be crucial [133].

Unfortunately, face images under different poses lose the feature correspondences because of missing facial regions, unequal dimensions and/or *region displacements*. Region displacement refers to the same facial region at different indices in feature vectors (see Fig.1.3). This example shows that even though we have used pixel intensities as features, pose difference led to a different modality. On the other hand, the case of text-image retrieval gave rise to a case where the two modalities represent entirely different concepts. It is this lack of harmony between representations that requires a common representation scheme to facilitate any meaningful relation between samples from different modalities.

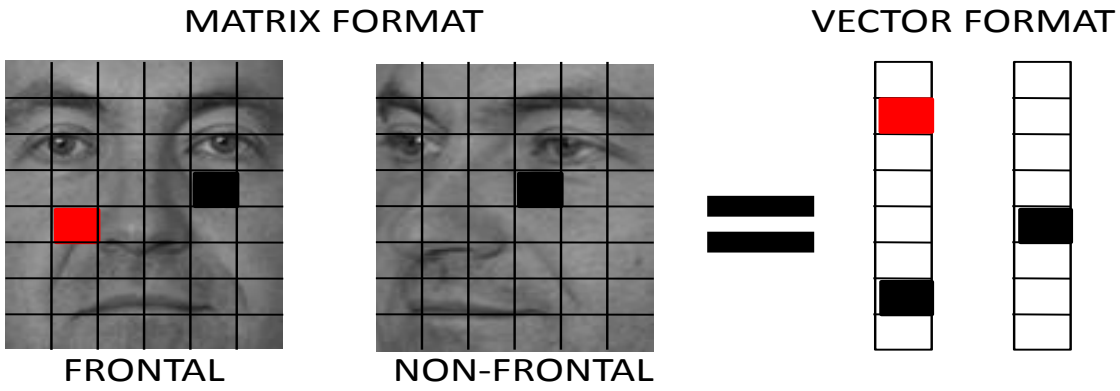


Figure 1.3: An example showing lack of correspondence due to missing regions and region displacement for pose variation. Black and red blocks indicate region displacement and missing region, respectively.

1.2 An overview of the Dissertation

This dissertation consists of two parts. The first part describes a novel neural network based approach for visual content extraction from images. In particular, a deep recursive neural network based semantic segmentation of real-life images has been proposed. Arguably, a pixel-wise dense segmentation mask is a rich form of visual content, because it can be used to understand the scene, objects and their relative positions, interactions between the objects and the world in the image. The proposed approach not only yields state-of-the-art results on benchmark datasets, but it is also orders of magnitude faster than the competing approaches.

In the second part we draw motivation from the common representation hypothesis and build mathematical models to extract content from multi-modal data in a form that affords cross-modal content matching. We attempt to explore and understand the shortcomings of previous approaches and build richer and more accurate models for some of the discussed problems. These problems were tackled using different approaches in the past. We, on the other hand, try to tackle these seemingly different problems using the same general idea of common representation and show impressive improvements in the current state-of-the-art. The success of our common representation approach on different problems validates the hypothesis and motivates us to explore richer models with more human-centric abilities. In particular, we develop a Partial Least Square based common representation for multi-modal face biometrics and extended it to a two-stage discriminative model to handle pose-errors. We formulated a common framework to extend any content extraction technique, such as Principal Component Analysis [82], Linear Discriminant Analysis [8, 125], Marginal Fisher Analysis [138] etc., to their multi-modal counterpart, given that the original problem can be formulated as an eigenvalue problem.

The rest of dissertation is organized as the following.

Chapter 2 provides a brief background on semantic segmentation, neural networks and some previous mathematical models to obtain common representation.

Chapter 3 presents the neural network model, Recursive Context Propagation Network or RCPN, for semantic segmentation.

Chapter 4 presents a PLS based common representation for multi-modal face recognition.

Chapter 5 extends the PLS based model to a two-stage discriminative architecture to handle small pose-errors. The adverse effect of pose-errors and the advantages of the proposed extension is also shown for larger datasets, such as FERET and MultiPIE.

Chapter 6 describes our supervised common representation model, Generalized Multiview Analysis or GMA, for pose and illumination invariant face recognition and text-image retrieval.

Chapter 7 summarizes the dissertation with conclusion and future directions.

Chapter 2

Background

This chapter provides a brief overview of some basic concepts required for understanding the material presented in this dissertation. We talk about 1) visual feature extraction with a focus on semantic segmentation, 2) deep neural networks and training and 3) a few popular techniques for learning a common representation from multi-modal data, in the order.

2.1 Visual feature extraction

Computer vision has been an active field of research for more than five decades. It has witnessed a colossal increment in the degree of sophistication and variety in terms of computer-vision based tasks. A principle requirement for almost every task is visual feature extraction that makes it one of the most active fields of research within computer-vision community. The large variation between different tasks is also reflected in the variety of the used visual features and the techniques to obtain them. The aforementioned variation between the visual features is so much so that there are several major areas of study solely dedicated to different feature extraction techniques. One such area is semantic segmentation owing to its challenging nature and wide-spread use with various tasks such as navigation, action recognition, medical images analysis and image understanding. A pixel-wise segmentation mask affords information such as objects in the images and their relative positions, the scene type in the image and interactions between animate and inanimate objects. This information is essential for a successful matching of an image to a verbal sentence that carries similar semantics. Although, there are other visual features, such as SIFT [77], HOG [22], GIST [85], Gabor features [136] and Fisher vectors [86], that can offer similar visual information, but the aforementioned applications of semantic segmentation along with its potential use in text-image matching motivates us to develop accurate and real-time algorithms for it.

2.1.1 Semantic Segmentation

Semantic segmentation aims at labeling each pixel of an input image as one of the required semantic categories. Fig. 2.1 illustrates an input image and its semantic segmentation mask. The immense variability in the appearance of objects, man-made and natural structures makes the problem very challenging. For example, in Fig. 2.1, it is required to correctly label the sand, sky and water regions despite very close visual similarity between them.

There are various approaches to obtain a semantic segmentation mask of a given image. Some approaches try to label individual pixels into semantic categories [44, 60, 114, 115]. Unfortunately, the per-pixel labeling approaches lead to a noisy labeling with lots of miss-classified small isolated pixels or group of pixels. The noisy per-pixel approaches can benefit by labeling at *super-pixel* level. Super-pixels are contiguous similar color or

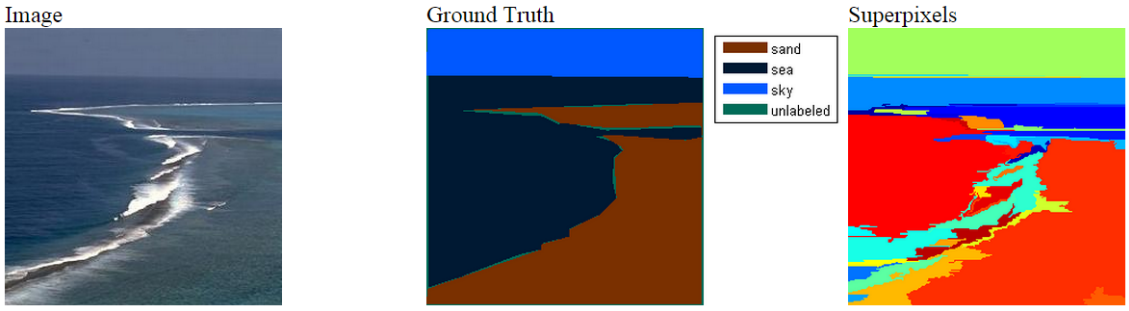


Figure 2.1: An example of a color image with its semantic segmentation mask and super-pixel segmentation. Different super-pixels are depicted by different colors.

texture regions of an image, please see Fig. 2.1 for an example. A few popular algorithms to obtain super-pixels from an input image are [30, 73, 2]. Some semantic segmentation approaches that work at super-pixel level are [93, 119, 79, 48]. These approaches assign the same label to each pixel within a super-pixel. Some semantic categories are particulary challenging due to large variation in their color, texture, shape or view-point such as human, car, bike and animals. These objects are first detected using object detectors and the detection bounding boxes are used to obtain a per-pixel segmentation of the object within the bounding box [45, 40, 4, 41].

2.2 Neural Networks and Related Concepts

A neural network maps an input to an output. Any artificial intelligence problem can be thought of as a mapping from an input to an output space, therefore, it can be modeled as a neural network. The input and output spaces depend on the problem in hand. For example, object classification takes an image as the input and outputs the classes of the present objects in the image. An excellent tutorial on neural networks and their use in artificial intelligence tasks is given in [10]. Here, we gloss over some of the key-concepts required to facilitate the understanding of the material in this dissertation, please refer to [10] for a detailed discussion of any of the following concepts.

Mathematically a neural network can be defined as -

$$\mathbf{y} = f(\mathbf{x}; \theta) \tag{2.1}$$

here, $\mathbf{y}$, $\mathbf{x}$ and θ are output, input and network parameters. Almost always, a neural network is broken down into a layer-wise structure with non-linear function sand-witched between the layers. Fully-connected [100], convolutional [21], recurrent [47, 104, 103] and recursive [89, 119] neural networks are the most popular neural networks among various possible neural architectures [10]. This dissertation does not deal with recurrent neural network, therefore, all the discussions from now onwards pertaining to neural network will be applicable to fully-connected, convolutional and recursive networks only.

Network parameters (θ) refer to the set of learnable parameters of each layer, commonly referred to as layer weights. The layer weights depend on the architecture type and the number of neurons present in the layers. For example, a fully-connected layer fc_l with 1000 input and 200 output neurons will have its weight parameter $W_l^{fc} \in \Re^{200\times1000}$ and a convolutional layer cn_l with a convolution kernel of size 5×5, 100 input and 200

output feature maps will have $W_l^{cn} \in \Re^{200\times100\times5\times5}$.

Capacity of a neural network refers to the possible complexity of the mapping function given a particular architecture and hyper-parameter setting such as type of non-linear function, number of layers and neurons within each layer. The capacity of a network always increases with an increase in the number of layers and neurons [49]. From the examples of fully-connected and convolutional networks we can see that the increase in the number of layers and neurons leads to an increase in the total number of network parameters as well.

Loss function takes the output of a neural network and the corresponding ground-truth and maps them to a scalar value proportional to the penalty incurred for that input under the current settings of θ -

$$\mathcal{L}(\mathbf{y}, \tilde{\mathbf{y}}; \theta) \in \{l : \mathbf{y} \times \tilde{\mathbf{y}} \rightarrow \Re\} \tag{2.2}$$

here, $\mathbf{y}$ and $\tilde{\mathbf{y}}$ are the network output and ground-truth, respectively.

Training data refers to a collection of inputs ($\mathbf{x}_i$) and the corresponding ground-truth labels ($\tilde{\mathbf{y}}_i$) suitable for the problem in hand. For example, object classification problem can use a training data of visual images that contain objects and a label for each image that indicates the presence and absence of each object category.

Parameter learning refers to the process of altering θ to bring down the cumulative loss function over the given training data. The learning process is commonly known as *back-propagation* due to the fact that gradients with respect to the network parameters is facilitated through reverse propagation of error from the final layer to the data input layer [100]. Typically, gradient descent or stochastic gradient descent with mini-batches of training data is used to bring down the loss function. The learning stops when the gradient of the parameters become vanishingly small. Naturally, we would like to obtain the global minimum of the loss function for the optimal performance. Unfortunately, due to the highly non-convex nature of the loss function the learning will inevitable stop at a **local minimum** solution and our best hope is to settle down in a useful local minimum.

Over-fitting refers to the scenario when the learned network can correctly predict the ground-truth for the training data inputs, but fails to correctly predict the outputs for the test data. Over-fitting may occur due to several reasons such as small training set, a training set that is not representative of the test set and unnecessary capacity that allows the network to fit the idiosyncracies of training data. Please note that the small training set and unnecessary network capacity are relative in their effect, for example, a network with large capacity can be made to avoid over-fitting, commonly referred to as *generalize*, by training it with a large amount of data and similarly, smaller datasets can be used for training with controlled network capacity.

Regularization refers to the set of techniques that can be used to avoid over-fitting while training a neural network. Common regularization techniques are - $l2$-penalty on the layer weights, data-augmentation [58], controlling the number of network parameters by parameter sharing [21, 89], unsupervised pre-training [26], supervised pre-training on large datasets [76, 54] etc.. While training neural network for complicated tasks it is a common practice to start with a large-capacity network without any regularization and keep on increasing the capacity with the degree of regularization to reach a point where an increase in capacity does not effect the performance either positively or adversely.

Dropout refers to a technique that has been proposed recently and it is found to be extremely effective for obtaining a better solution [120]. It randomly omits a fraction

of neurons from the computation graph of the neural network during a forward-backward propagation cycle. The exact effect of dropout is not yet clear, but it consistently improves the performance for almost every deep neural network based approach.

2.3 Common Representation Extraction

As stated earlier, we are not the first one to explore the common representation hypothesis. This idea goes back to at least 1933 when it first appeared in the seminal paper of Hotelling in the form of Canonical Correlation Analysis [38]. Since then, there has been a vast amount of work on building models to learn a common representation from different modalities. The large volume of available literature forbids an exhaustive review, therefore, we will only discuss the approaches that were either used in our study or similar to our work.

A popular approach for finding a common representation for cross-modal content matching is to learn linear/non-linear projection directions in each modality to project samples from different modalities into a common subspace. The projection directions are obtained as solutions of different optimization problems. The two lines of optimization problem come from subspace and probabilistic approaches. Canonical Correlation Analysis (CCA), Bilinear Model (BLM) and Partial Least Squares (PLS) are representatives of subspace approaches and Tied Factor Analysis (TFA), shared private model etc. represent probabilistic approaches. In the discussion that follows, we will use face recognition as the working example. Therefore, the task would be to find the identity of the person, content could be any feature vector that is useful for the task and modalities would be different representations of the content.

2.3.1 Bilinear Model

Tannenbaum and Freeman [128] proposed a bilinear model (BLM) for separating *modality* and *content*. They suggested different methods for learning BLMs and using them in a variety of tasks, such as identifying the modality of a new image with unfamiliar content, or generating novel images based on separate examples of the modality and content. However, their approach also suggests that their content-modality models can be used to obtain a modality-invariant content representation that can be used for relating a sample in different modalities. Following their asymmetric model, they concatenate the i^{th} subject's images under M different poses ($\mathbf{y}_m^i : m = 1, 2, \ldots M$) to make a long vector $\mathbf{y}^i$ and construct matrix Y having columns as $\mathbf{y}^i$ with $i = \{1, 2, \ldots N = \#subjects\}$ such that:

$$Y = \begin{pmatrix} \mathbf{y}_1^1 & \mathbf{y}_1^2 & \cdots & \mathbf{y}_1^N \\ \mathbf{y}_2^1 & \mathbf{y}_2^2 & \cdots & \mathbf{y}_2^N \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_M^1 & \mathbf{y}_M^2 & \cdots & \mathbf{y}_M^N \end{pmatrix} = \begin{pmatrix} \mathbf{y}^1 & \mathbf{y}^2 & \ldots & \mathbf{y}^N \end{pmatrix} \tag{2.3}$$

Modality matrices A_m which can be thought of as different representation schemes and can be obtained by decomposing the matrix Y using SVD as -

$$Y = USV^T = (US)V^T = (A)B \tag{2.4}$$

A can be partitioned $A^T = \begin{pmatrix} A_1^T & A_2^T & \ldots & A_M^T \end{pmatrix}$ to give different representation schemes A_m's where m represents different poses.

2.3.2 Canonical correlational analysis

Canonical Correlational Cnalysis or CCA is a technique that learns a set of M different projection directions from a set of observed *content* under M different *modalities*. The projections of different *modalities* of a particular *content* are maximally correlated in the projected space. Hence, CCA can be used to learn a common intermediate subspace in which projections of different pose images of the same subject will be highly correlated and recognition can be done on the basis of the correlation score. Given a set of face images of N different subjects under M different poses, CCA learns a set of K dimensional subspaces $W_m = \{\mathbf{w}_m^k : \mathbf{w}_m^k \in \Re^{Dm}; k = 1, 2, \dots K\}$ for $m = 1, 2, \dots M$ such that [39]:

$$\begin{pmatrix} C_{11} & C_{12} & \dots & C_{1M} \\ C_{21} & C_{22} & \dots & C_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ C_{M1} & C_{M2} & \dots & C_{MM} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1^k \\ \mathbf{w}_2^k \\ \vdots \\ \mathbf{w}_M^k \end{pmatrix} = (1+\lambda^k) \begin{pmatrix} C_{11} & 0 & \dots & 0 \\ 0 & C_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & C_{MM} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1^k \\ \mathbf{w}_2^k \\ \vdots \\ \mathbf{w}_M^k \end{pmatrix}$$
$$CW = W(I + \Lambda) \tag{2.5}$$

where, Dm is the feature dimension of the mth modality, $C_{ij} = \frac{1}{N} Y_i (Y_j)^T$ and Λ is a diagonal matrix of eigen-values λ^k, N is the number of training subjects, each pose is a modality and Y_i is defined in the previous sub-section. Equation (2.5) is a generalized eigenvalue problem which can be solved using any standard eigensolver. The columns of the projector matrices W_m will span a linear subspace in modality m. So, when the modalities are different poses, we get a set of vectors spanning a linear subspace in each pose.

2.3.3 Partial Least Squares

Partial Least Square analysis [113, 98, 1, 14], also known as PLS, is a regression model that differs from Ordinary Least Square regression by first projecting the regressors (input) and responses (output) onto a low dimensional latent linear subspace. The PLS projectors try to maximize the covariance between latent scores of regressors and responses. Hence, we can use PLS to obtain common representation for two different poses in the same way as BLM and CCA.

There are several variants of PLS analysis based on the objective function and related constraints to learn the latent space, see [14] for details on different PLS algorithms. In this paper, we have used the factor model assumption given in [14, 98] to develop intuitions and a variant of NIPALS given in [1] to learn the projection directions.

Following the same conventions as for BLM and CCA, Y_p represents a matrix containing face images in pose p as its columns. PLS greedily finds vectors $\mathbf{w}_p$ and $\mathbf{w}_q$ such that -

$$\begin{aligned} [\mathbf{w}_p, \mathbf{w}_q] &= \underset{\mathbf{w}_p, \mathbf{w}_q}{\operatorname{argmax}} \left(cov[Y_p^T \mathbf{w}_p, Y_q^T \mathbf{w}_q]^2 \right) \\ s.t. \quad & \|\mathbf{w}_p\| = \|\mathbf{w}_q\| = 1 \end{aligned} \tag{2.6}$$

MATLAB code of the NIPALS [1] based variant for learning the common latent space is given below.

```
function [W,Z] = PLS_bases(X,Y,nfactor)
```

```
% INPUT PARAMETERS
% X and Y both are supplied in a form where each column contains one sample
% nfactor - # desired PLS fatcors

% OUTPUT PARAMETERS
% W - the projection directions for X as columns
% Z - Projection directions for Y as columns
XD = size(X,1); % X-dimension
YD = size(Y,1); % Y- dimension

% Input check
if nargin < 3
    print ('Not enough input arguments probably missing nfactor')
    return;
end

% Number of samples
nx = size(X,2);
ny = size(Y,2);
if nx == ny
    n = nx;
else
    print ('The number of samples in X and Y are different');
    return;
end

% make them as row vectors now

X = X';
Y = Y';

% Initialisation of some matrices

W = zeros(XD,nfactor);
A = X'*Y;
M_ = X'*X;
C = eye(XD);
P = zeros(XD,nfactor);
Z = zeros(YD,nfactor);

for i = 1:nfactor

    [dumm d q] = svds(A,1);
    w = C*(A*q);
    w = w/norm(w);
    W(:,i) = w;
    p = M_*w;
```

```
    c = w'*p;
    p = p/c;
    P(:,i) = p;
    q = A'*(w/c);
    Z(:,i) = q;
    A = A - (c*p)*q';
    M_ = M_ - (c*p)*p';
    C = C - w*p';

end
```

2.3.4 Probabilistic common representation

These approaches use generative models to explain the data. Generally, they assume a common latent space, usually Gaussian for computational simplification, and use linear transformations to map the points in the latent space to the data points. Different architectures of connections give rise to different models with different properties. Some popular models are Tied Factor Analysis (TFA) [91], Probabilistic Linear Discriminant Analysis (PLDA) etc.. A common feature of all these approaches is the use of the EM algorithm for inference which is prone to local minima. Therefore, these approaches cannot guarantee convergence to the optimal solution. We do not discuss these approaches in detail because we are not going to use them in this dissertation.

Chapter 3

Deep Recursive Hierarchical Scene Parsing

This chapter introduces a novel recursive neural network architecture, referred to as Recursive Context Propagation Networks (RCPN), for semantic segmentation of images. RCPN first maps the local visual features into a semantic space followed by a bottom-up aggregation of local information into a global representation of the entire image. Then a top-down propagation of the aggregated information takes place that enhances the contextual information of each local feature. Therefore, the information from every location in the image is propagated to every other location. RCPN is further analyzed and modified accordingly to improve the model. The presence of bypass error paths, in the computation graph of RCPN, that can hinder contextual propagation is discovered by analyzing the temporal gradient strength during training. The classification loss of the internal nodes of the random parse trees in the original RCPN model is added to the loss function to tackle the problem that leads to Pure-node RCPN (PN-RCPN). Secondly, a novel tree-MRF on the parse tree nodes is used to model the hierarchical dependency present in the output, leading to Tree-MRF RCPN (TM-RCPN). Experimental results on Stanford background, SIFT Flow and Daimler Urban datasets show that the proposed methods outperform previous approaches in terms of accuracy for semantic segmentation. Most notably, RCPN and PN-RCPN are orders of magnitude faster than previous methods, except for Daimler dataset, and take only 0.07 seconds on a GPU for pixel-wise labeling of a 256×256 image starting from raw RGB pixel values, given the super-pixel mask that takes an additional 0.3 seconds using an off-the-shelf implementation. This work is the result of a collaboration with Dr. Oncel Tuzel and Dr. Ming-Yu Liu, a part of this work was completed during my internship at Mitsubishi Electric Research Lab Cambridge.

3.1 Motivation and Introduction

As discussed earlier in Sec. 2.1.1, semantic segmentation aims at getting pixel-wise dense labeling of an image in terms of semantic concepts such as tree, road, sky, water, foreground objects etc. The pixel-wise dense semantic mask for an image is a very rich form of visual information that can be leveraged for a variety of complicated visual tasks such as navigation, scene understanding, robotics, and medical image analysis. The versatility and usefulness of dense semantic mask calls for highly accurate and real-time algorithms to obtain it from the input images. Unfortunately, the rich diversity in the appearance of even simple concepts (sky, water, grass) makes semantic segmentation very challenging. Surprisingly, human performance is almost close to perfect on this task. This striking difference of performance has led to a heated field of research in the vision community. Past experiences and recent research [131, 81, 80] have conclusively established that the ability of humans to utilize the information from the entire image is the main reason behind the large performance gap. Interestingly, [81, 80] have shown that human performance in labeling a small local region (super-pixel) is worse than a computer when both are looking at only that region of the image.

Mathematically, semantic segmentation can be framed as a mapping from a set of nodes arranged on a 2D grid (pixels) to the semantic categories. Typically semantic segmentation can be broken down into two steps - feature extraction and inference. Feature extraction involves the retrieval of descriptive information for semantic labeling under varying illumination and view-point conditions. These features are generally color, texture or gradient based and extracted from a local or large patch around each pixel. The inference step consists of predicting the labels of the pixels using the extracted features. Taking a cue from human performance with and without contextual information, previous works have developed increasingly sophisticated inference algorithm to utilize the information from the entire image via different algorithms. Markov Random Fields (MRFs) [68], Conditional Random Fields (CRFs) [62, 88] and Structured Support Vector Machines (SVMs) [132] are among the most successful and widely used algorithms for inference.

The basic philosophy behind the aforementioned inference algorithms is to model the distribution of label nodes according to a *hypothesized* image formation process. The interaction between the label nodes and the observed image is facilitated through binary or higher-order interaction potential functions. The difference between these approaches is mainly in terms of 1) image formation process, 2) type of potential function used and 3) parameter estimation and test-time inference. MRF based approaches model the joint distribution of the node labels via binary or higher order potential functions on the 2D label grid. These potential functions are hand-designed to conform to the common image models, such as smoothness of the label field. The unary potentials are employed through local visual features for each node. CRFs model the joint distribution of the node labels *given* the observations and can include higher order potentials in addition to the unary potentials. Higher order potentials allow these models to represent complex dependencies between the node labels, which is important for structured prediction tasks. On the downside, except for a few exceptions such as non-loopy models, the inference algorithms for these models require solving a non-submodular discrete optimization problems, which can be only approximately solved and are time consuming. Moreover, parameter learning procedures that are tractable usually limit the form of the potential functions to simple forms such as linear models.

From the discussion so far, it is clear that utilization of information from the entire image can result in more accurate semantic segmentation, but the complicated inference step introduces a critical trade-off between accuracy and efficiency. I seek motivation from these facts and develop a *computationally efficient* feed-forward neural network that utilizes the information from the entire image while labeling any region of the image.

3.2 Overview of Proposed Approach

This section presents the intuition and outlines the proposed approach for semantic segmentation. The semantic segmentation task is modeled as a learnable mapping from the set of all pixels in an image $\mathbf{I}$ to the corresponding label image $\mathbf{Y}$. There are several design considerations for a desired mapping -

- Real-time evaluation for robotics, urban scene understanding, etc..
- It should utilize the entire image such that every location can potentially influence the labeling of every other location for greater accuracy. Please refer to Fig. 3.1 and

observe that the white boat in isolation can be confused with a white building, but the presence of water removes this confusion and we can tell that its a boat.

- Its parameters should be learned from the training data to remove the need for domain expertise and to minimize the need for human-input.
- It should scale to different image sizes.
- It must generalize to unseen images that requires limiting the capacity of the mapping while still utilizing the entire image information at once. For example, a simple fully-connected-linear mapping from $\mathbf{I}$ to $\mathbf{Y}$ would require 4 Trillion parameters for an image of size 256×256, but it will fail to generalize under practical conditions of limited training data. Please refer to Sec. 2.2 for a better understanding.

Considering the requirements discussed above, the mapping is designed as a single feed-forward neural network with carefully controlled capacity by parameter sharing. All the network parameters are learned from the data and the feed-forward structure allows very fast, almost real-time, inference. The proposed network can be functionally partitioned into two sub-networks: local feature extraction and recursive context propagation.

As the name implies, local-feature extraction refers to the extraction of pixel- or region-wise visual features for semantic labeling. The multi-scale convolution neural network (Multi-CNN) proposed in [28] is used to get pixel-wise features. A convolution structure with shared parameters brings down the number of parameters for local feature extraction, thereby offers better generalization.

The main contribution of this chapter is a novel recursive context propagation network (RCPN) and its derivatives, which, starting from the local features, recursively aggregate contextual information from local neighborhoods up to the entire image and then disseminate the aggregated information back to individual local features for better semantic classification. RCPN is a recursive neural network with shared parameters through the parse tree hierarchy. A conceptual illustration of such networks is given in Figure 3.1. The scene consists of three segments corresponding to a boat, a tree and a water/sky region. The nodes of the graph (formed by a binary parse tree and its inversion) represent a semantic description of the segments. The distributions on the left are probable label distributions for the segments, based on their appearance. Initially (at the bottom), the boat can be confused with a white building while looking only at the bottom-left segment. The RCPN recursively combines two segment descriptions and produces the semantic description of the combined segment. Therefore, as the tree is combined with the boat, the belief that the combined segment includes a building increased since usually they appear together in the images. Similarly, after we merge the water/sky segment description with the boat-tree segment description, the probability of the boat increased since the simultaneous occurrence of water and building is rare. The middle node in the graph (root node of the segmentation tree) corresponds to the semantic description of the entire image. After all the segment descriptions are merged into a single holistic description of the entire image, this information is propagated to the local regions. This is achieved by recursive updates of the semantic descriptions of the segments given the descriptions of their parent segments. Finally, *contextually enhanced* descriptions of the leaf nodes are used to label the segments. Note that RCPN only uses segment descriptions and not the illustrative-only label distributions shown in the figure.

RCPN is influenced by Socher et al.'s work [119] that learns a non-linear mapping from feature space to a semantic space, termed semantic mapping. In [119], the semantic

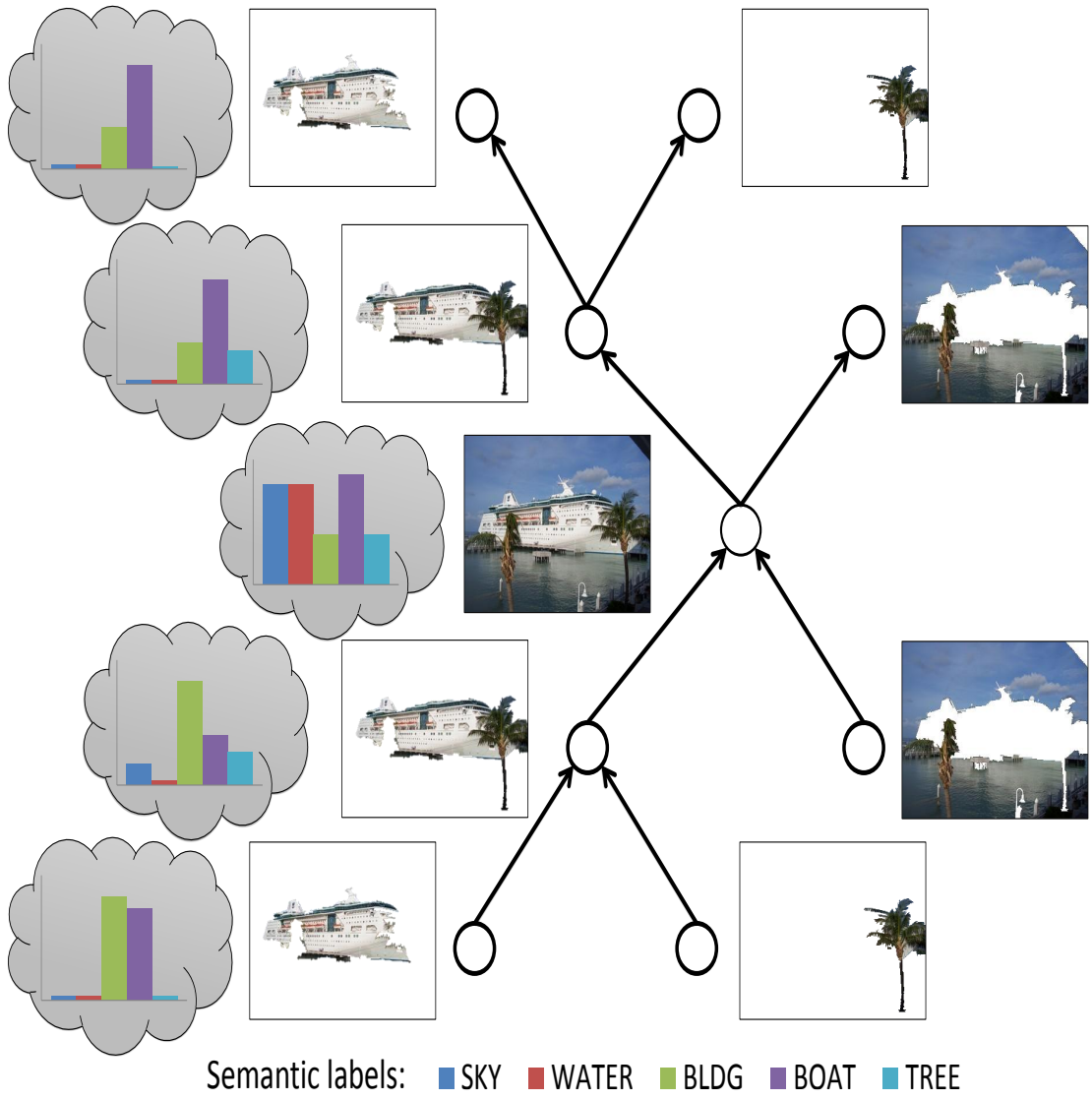


Figure 3.1: Conceptual illustration of the proposed RCPN architecture for semantic segmentation. RCPN recursively aggregates contextual information from local neighborhoods to the entire image and then disseminates global context information back to individual local features. In this example, starting from a confusion between boat and building, the propagated context information helps resolve the confusion by using the feature of the water segment. Please note that the probability distributions are only meant to convey the confidence of presence/absence of a particular class in the RCPN hierarchy for the associated image-region.

space is learned by optimizing a structure prediction cost on the ground-truth parse trees of training images or sentences. Next, a classifier is learned on the semantic mappings of the individual local features from the training images. At test time, individual local features are projected to the semantic space using the learned semantic mapping followed by classification. Therefore, only the information contained in an individual feature is used for labeling. In contrast, RCPN uses recursive bottom-top-bottom paths on randomly generated parse trees to propagate contextual information from local regions to all other regions in the image. Therefore, it is expected to do better, please see experiments section for detailed comparison.

The major advantages of the RCPN family of networks are -

- **Scalability** - RCPNs are a combination of CNN and recursive neural network and the entire pipeline can be trained without using any human-designed features. In addition, convolution+recursive structure allows scaling to arbitrary image sizes while still utilizing the entire image content at once.

- **Performance** - RCPNs achieve state-of-the-art segmentation accuracy on three important benchmarks while being an order of magnitude faster than the existing methods. This enormous speed-up is possible due to the feed-forward operations only. For instance, it takes only **0.07** seconds on GPU and **0.8** seconds on CPU for pixel-wise semantic segmentation of a 256×256 image, with a given super-segmentation mask, that can be computed using an off-the-shelf algorithm within 0.3 second.

- **Modularity** - Proposed RCPN modules can be used in conjunction with pre-computed features to propagate context information through the structure of an image (see experiments section) and potentially other structured prediction tasks.

- **Hierarchical scene understanding** - The pure-node variant of RCPN, referred to as PN-RCPN, provides an opportunity to parse the image at multiple resolutions through the parse tree hierarchy. The label distributions at various resolutions of image regions are used to enforce spatial smoothness and local-global label consistency through an MRF model on the parse tree that leads to Tree-MRF RCPN or TM-RCPN.

3.3 Semantic Segmentation Architecture

This section describes the proposed semantic segmentation architecture and discusses the design choices for practical considerations. An illustration of this architecture is shown in Figure 3.2. The input image is fed to a CNN, F_{CNN}, which extracts local features per pixel. Then a super-pixel tessellation of the input image and average pooling of the local features within the same super-pixel is carried out to obtain visual features for each super-pixel. Lastly, one of the proposed RCPN derivatives is used to obtain the final labels of each super-pixels.

3.3.1 Local feature extraction

End-to-end trainability is a desired requirement for our pipeline, therefore, we resort to a learning based approach for feature extraction. Multi-scale convolution neural

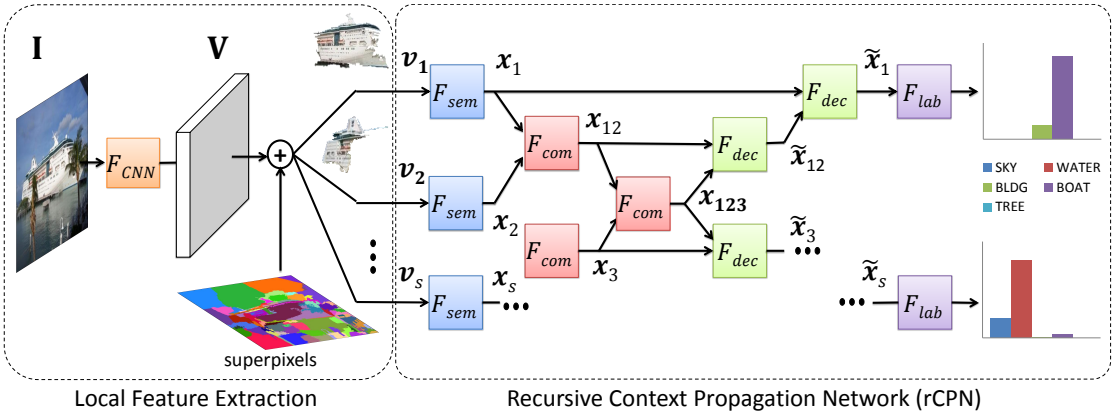


Figure 3.2: Overview of semantic scene labeling architecture

network (Multi-scale CNN) [28] based pixel-wise features are used for per-pixel visual feature extraction. The process outlined in [112] was used with the same CNN structure and similar preprocessing (subtracting 0.5 from each channel at each pixel location in the RGB color space) at 3 different scales (1,1/2 and 1/4) to obtain the visual features. The CNN architecture has three convolutional stages with $8 \times 8 \times 16$ *conv* $\rightarrow 2 \times 2$ *maxpool* $\rightarrow$ $7 \times 7 \times 64$ *conv* $\rightarrow 2 \times 2$ *maxpool* $\rightarrow 7 \times 7 \times 256$ *conv* configuration. Each max-pooling operation is non-overlapping. Therefore, every image scale has 256 dimensional features for each pixel. Multi-scale CNN obtains per-pixel visual features from a patch around each of the pixels. The size of the patch depends on the CNN parameters and the scales. Features extracted at different scales give rise to a different field of view (FOV) for each pixel. For example, using the aforementioned architecture with all the three scales results in 188×188 FOV.

Note that the 768 dimensional concatenated output feature map is still 1/4th of the height and width of the input image due to the two max-pooling operations. To obtain the input size per-pixel feature map we have two possible options

- **Slow:** Shift the input image by one pixel on a 4×4 grid to get 16 output feature maps that can be combined to get the full-resolution image.
- **Fast:** Scale-up each feature map by a factor of 4 in height and width using Bilinear interpolation.

Empirically, the latter was found out to be equally accurate as the former.

3.3.1.1 Super-pixel representation

Although it is possible to do per-pixel classification using the RCPN, learning such a model would be computationally intensive and the resulting network would be prohibitively deep to propagate the gradients efficiently due to recursion. To reduce the complexity, the super-pixel segmentation algorithm of [73] is utilized, which provides the desired number of super-pixels per image. We average pool the per-pixel local features within the same super-pixel and retrieve s local features, $\{\mathbf{v}_i\}_{i=1...s}$, one per super-pixel. During training however, 5 different sets of 5 random pixels in a super-pixel are averaged to obtain 5 different visual features per super-pixel, it is done to expand the training data to avoid over-fitting.

3.3.2 Inference

This step corresponds to obtaining the final semantic labels for each pixel (or super-pixel). The following sections describe the basic RCPN model and its derivatives as inference modules for semantic segmentation.

3.4 Recursive Context Propagation Network

The basic RCPN architecture consists of four neural networks -

- F_{sem} maps local features to the semantic space in which the local information can be propagated to other segments.
- F_{com} recursively aggregates local information from smaller segments to larger segments through a parse tree hierarchy to capture a holistic description of the image.
- F_{dec} recursively disseminates the holistic description to smaller segments using the same parse tree.
- F_{lab} classifies the super-pixels utilizing the contextually enhanced features.

3.4.1 Parse tree synthesis

Both for training and inference, the binary parse trees, used for propagating information through the network, are synthesized randomly. The parse trees are obtained by randomly combining two adjacent super-pixels. The synthesis algorithm favors roughly balanced parse trees by greedily selecting sub-trees with smaller heights at random. The parse trees are only used as computation paths to propagate the contextual information throughout the image. Therefore, it is not needed that the parse trees represent an accurate hierarchical segmentation of the image, unlike [63, 119]. The MATLAB code for generating roughly balanced binary parse trees from the adjacency matrix of the super-pixels is given below -

---

```
function parents =  BalancedRandomTree(adjMatrix)
% adjMatrix is the adjacency matrix of the super-pixel neighbor graph
% parents is the output tree structure
numNodes = size(adjMatrix,1);
allNodeIndices = [1:2*numNodes-1];
nodeDepths = zeros(1,2*numNodes-1);
parents = zeros(1,2*numNodes-1);
extendedAdjMatrix = sparse(2*numNodes-1,2*numNodes-1);
extendedAdjMatrix(1:numNodes,1:numNodes) = adjMatrix;
nCurrentNodes = numNodes;

while nCurrentNodes < 2*numNodes-1
    for currentDepth=0:max(nodeDepths)
        tmpExtendedAdjMatrix = extendedAdjMatrix;
        bigDepthIndices = find(nodeDepths > currentDepth);
        if (~isempty(bigDepthIndices))
```

```
            tmpExtendedAdjMatrix(bigDepthIndices,:) = 0;
            tmpExtendedAdjMatrix(:,bigDepthIndices) = 0;
        end
        indices = find(tmpExtendedAdjMatrix(1:nCurrentNodes,1:nCurrentNodes));
        if (length(indices) > 0)
            randIndex = ceil(rand() * length(indices));
            index = indices(randIndex);
            [firstNode secondNode] = ind2sub([nCurrentNodes nCurrentNodes], index);
            parents(firstNode) = nCurrentNodes+1;
            parents(secondNode) = nCurrentNodes+1;
            extendedAdjMatrix(nCurrentNodes+1,:) = extendedAdjMatrix(firstNode,:)...
            | extendedAdjMatrix(secondNode,:);
            extendedAdjMatrix(:,nCurrentNodes+1) = extendedAdjMatrix...
            (nCurrentNodes+1,:)';
            extendedAdjMatrix(firstNode,:) = 0;
            extendedAdjMatrix(:,firstNode) = 0;
            extendedAdjMatrix(secondNode,:) = 0;
            extendedAdjMatrix(:,secondNode) = 0;
            nodeDepths(nCurrentNodes+1) = max(nodeDepths(firstNode), ...
             nodeDepths(secondNode)) + 1;
            nCurrentNodes = nCurrentNodes+1;
            break;
        end
    end
end
```

The code takes the adjacency matrix of the super-pixel graph as input. Initially all the nodes (super-pixels) are detached and have depth 1. We randomly select two nodes among the lowest depth nodes that are adjacent, and combine them into a merged-node. The depth of the merged-node is set to the max depth of the two child nodes plus one and its adjacency matrix is set to the union of the neighbors of the two child nodes. The two child nodes are then removed from the graph. This process is repeated until there remains a single merged-node corresponding to the union of all the nodes (entire image). With this algorithm, the expected depth of the parse tree becomes $O(log_2 S)$, S is the number of super-pixels, except for the degenerate adjacency matrices.

3.4.2 Semantic mapping network

The semantic mapping network is a feed-forward neural network that maps the d_v dimensional local features $\mathbf{v}_i$ to the d_s dimensional semantic vector space

$$\mathbf{x}_i = F_{sem}(\mathbf{v}_i; \mathbf{W}_s). \tag{3.1}$$

where $\mathbf{W}_s : \Re^{d_v} \rightarrow \Re^{d_s}$ is the model parameter for the semantic module. The aim of the semantic features is to capture a joint representation of the local features and the context, and being able to propagate this information through a parse tree hierarchy to other super-pixels. Please note that $\mathbf{W}_s$ will be a $d_s \times d_v$ matrix for a single layer semantic mapper and a cell of matrices with appropriate sizes for multi-layer semantic mapper.

3.4.3 Combiner network

The combiner network is a recursive neural network that maps the concatenation of semantic features of two child nodes, each associated with either a super-pixel or a merged-region, in the parse tree to obtain the semantic feature of the parent node (combination of the two child nodes), associated with a merged-region

$$\mathbf{x}_{i,j} = F_{com}([\mathbf{x}_i, \mathbf{x}_j]; \mathbf{W}_c). \tag{3.2}$$

where $\mathbf{W}_c : \Re^{2d_s+1} \rightarrow \Re^{d_s}$ is the model parameter for the combiner network. Intuitively, the combiner network attempts to aggregate the semantic content of the children nodes such that the parent node becomes representative of its children. The information is recursively aggregated bottom-up from super-pixels to the root node through the parse tree. The semantic features of the root node correspond to the holistic description of the entire image.

3.4.4 Decombiner network

The decombiner network is a recursive neural network that disseminates the context information from the parent nodes to the children nodes throughout the parse tree hierarchy. This network maps the semantic features of the child node and its parent to the contextually enhanced feature of the child node

$$\tilde{\mathbf{x}}_i = F_{dec}([\tilde{\mathbf{x}}_{i,j}, \mathbf{x}_i]; \mathbf{W}_d). \tag{3.3}$$

where $\mathbf{W}_d : \Re^{2d_s+1} \rightarrow \Re^{d_s}$ is the model parameter for the decombiner network. The dissemination of the information content of the entire image, as the root feature, starts from the root and recursively propagates in a top-down manner till it reaches the super-pixel features, therefore, it is expected that every super-pixel feature contains the contextual information aggregated from the entire image. Therefore, it is influenced by every other super-pixel in the image.

3.4.5 Labeler network

The labeler network is the final feed forward network that maps the context enhanced semantic features ($\tilde{\mathbf{x}}_i$) of each super-pixel to the C dimensional label vector $\mathbf{y}_i$, C is the number of semantic categories.

$$\mathbf{y}_i = F_{lab}(\tilde{\mathbf{x}}_i; \mathbf{W}_l). \tag{3.4}$$

where $\mathbf{W}_l : \mathcal{R}^{d_s+1} \rightarrow \mathcal{R}^C$ is the model parameter for the labeler module. Contextually enhanced features contain both local and global context information, thereby leading to better classification.

Together, all the parameters of RCPN are denoted as $\mathbf{W}_{rcpn} = \{\mathbf{W}_s, \mathbf{W}_c, \mathbf{W}_d, \mathbf{W}_l\}$. Let's assume there are S super-pixels in an image I and denote a set of R random parse trees of I as $\mathcal{T}$. Then, the loss function for I is

$$\mathcal{L}(I) = \frac{1}{RS} \sum_{r=1}^{R} \sum_{s=1}^{S_i} L(\mathbf{y}_{r,s}, t_s; \mathcal{T}_r, W_{rcpn}) \tag{3.5}$$

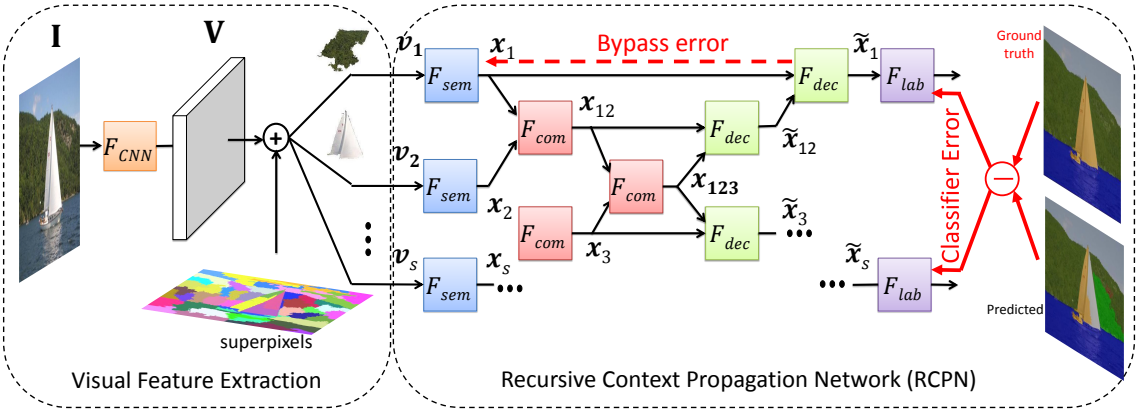


Figure 3.3: Learning schematic of RCPN with the input as the raw image and label as the semantic mask of the input image.

here, $\mathbf{y}_{r,s}$ is the predicted class-probability vector and t_s is the ground-truth label for the s^{th} super-pixel for random parse tree $\mathcal{T}_r$ and $L(\mathbf{y}_s, t)$ is the cross-entropy loss function. Network parameters, W_{rcpn}, are learned by minimizing $\mathcal{L}(I)$ for all the images in the training data.

3.4.6 Side information

It is possible to input information to the recursive networks not only at the leaf nodes but also at any level of the parse tree. The side information can encode the static knowledge about the parse tree nodes and it is not a result of neural computations through the tree. The implementations in this chapter used average x and y locations of the nodes and their sizes as the side information.

3.5 Learning

The proposed segmentation architecture is a feed-forward neural network that can be fully trained using training data. However, the recursion makes the depth of the neural network too deep for an efficient joint training. Therefore, we first learn the CNN parameters ($\mathbf{W}_{cnn}$) using the raw image and the ground truth segmentation labels. The trained CNN model is used to extract per-pixel features that are used to obtain super-pixel features (see Sec. 3.3.1.1 for details) followed by training of RCPN ($\mathbf{W}_{rcpn}$) to predict the ground truth super-pixel labels. The schematic of learning process is shown in the Fig. 3.3.

3.5.1 Local feature extractor

Feature extractor CNN (F_{cnn}) is trained on a GPU using a publicly available implementation CAFFE [52]. In order to avoid over-fitting we used data augmentation and dropout [58, 120]. All the training images were flipped horizontally to get twice the original images and we also used 1-pixel shifted input image to increase the dataset by an additional factor of two. We used dropout in the last layer with dropout ratio equal to 0.5. Standard back-propagation for CNN is used with stochastic gradient descent update scheme on mini-batches of 6 images, with weight decay ($\lambda = 5 \times 10^{-5}$) and momentum ($\mu = 0.9$). It typically took 6-8 hours of training on a GPU as compared to 3-5 days

training on a CPU as reported in [28]. We found that simply using RGB images with ReLU units and dropout gave slightly better pixel-wise accuracy as compared to [28].

3.5.2 RCPN parameter learning

RCPN parameters are trained using back-propagation through structure [33], which back-propagates the error through the parse tree, from F_{lab} to F_{sem}. The basic idea is to split the error message at each node and propagate it to the children nodes. Limited memory BFGS [72] with line-search is used for parameter updates using publicly available implementation [1]. As explained earlier in Sec. 3.3.1.1, we generate 5 sets of features of each super-pixel and used a different random parse tree for each set of random feature, thus we increased our training data by a factor of 5. It typically took 600 to 1000 iterations for complete training.

3.6 Pure-node RCPN

In this section, the RCPN model is studied leading to a discovery of potential problems with parameter learning. Useful modifications to the learning and the model are also proposed to tackle the learning problems. Especially, it is shown that the direct path from the semantic mapper to the labeler gives rise to bypass errors that can cause RCPN to bypass the combiner and decombiner assembly. This can cause back-propagation to reduce RCPN to a simple multi-layer neural network for each super-pixel. A simple remedy is also proposed to tackle this issue by adding the classification loss of those internal nodes of the random parse trees that correspond to a single semantic category, referred to as pure-nodes, to the original RCPN loss function. This serves the following purposes -

- It provides more labels for training, which results in better generalization.
- It encourages stronger gradients deep in the network.
- It explicitly forces the combiner to learn meaningful combinations, because the internal node mis-classifications are penalized.
- Lastly, it tackles the problem of bypass errors, resulting in better use of contextual information.

3.6.1 RCPN analysis and pure-node inclusion

Here we propose a model that will handle bypass errors. At the same time, this model solves a problem of gradient attenuation, increases the training data and forces the combiner to learn meaningful combinations of features.

For ease of understanding all our discussions will be limited to 1-layer modules. This result in each of the $\mathbf{W}_s$, $\mathbf{W}_c$, $\mathbf{W}_d$ and $\mathbf{W}_l$ as matrices, denoted as W_s, W_c, W_d and W_l, respectively. Like most deep networks, RCPN also suffers from vanishing gradients for the lower layers. This stems from the vanishing error signal, because the gradient ($\mathbf{g}_n$) for the n^{th} layer depends on the error signal ($\mathbf{e}_{n+1}$) from the layer above -

$$\mathbf{g}_n = \mathbf{e}_{n+1}\mathbf{z}_n^T \tag{3.6}$$

[1] http://www.di.ens.fr/ mschmidt/Software/minFunc.html

here, $\mathbf{z}_n$ is the input to the n^{th} layer. For RCPN, vanishing gradients are more of a problem because of very deep parse trees due to recursion. For instance, a 100 super-pixel image will lead to a minimum of $(log_2(100) \times 2 + 2 > 14)$ layers under the strong assumption of perfectly balanced binary parse trees. In practice, we can only create roughly balanced binary trees that often lead to ~ 30 layers.

We show that the internal nodes of the parse tree can be used to alleviate these problems. Each node in the parse tree corresponds to a connected region in the image. The leaf nodes correspond to the initial super-pixels and the internal nodes correspond to the merger of two or more connected regions, referred to as merged-region. We use the term *pure nodes* to refer to the internal nodes of the parse tree associated with the merger of two or more regions of the same semantic category. Therefore, the merged-regions corresponding to the pure nodes can serve as additional labeled samples during training. We empirically found that roughly 65% of all the internal nodes are pure-nodes for all three datasets. We include the classification loss of the pure-nodes in the loss function (Eqn. 3.5) for training and refer to the new procedure as *pure-node RCPN* or PN-RCPN for short. The classification loss, $\mathcal{L}^p(I)$, now becomes -

$$\mathcal{L}^p(I) = \mathcal{L}(I) + \frac{1}{\sum P_r} \sum_{r=1}^{R} \sum_{p=1}^{P_r} L(\mathbf{y}_{r,p}, t_{r,p}; \mathcal{T}_r, W_{rcpn}) \tag{3.7}$$

here, P_r is the number of pure-nodes for the r^{th} random parse tree $\mathcal{T}_r$ and subscripts (r, p) map to the p^{th} pure-node for the r^{th} random parse tree. Note that different parse trees for the same image can have different pure nodes.

In order to understand the benefits of PN-RCPN and contrast it with RCPN, we make use of an illustrative example depicted with the help of Fig. 3.4. The left-half of a random parse tree for an image I with 5 super-pixels is shown in Fig. 3.4. The figure also contains various variables involved during one forward-backward propagation through RCPN (Fig. 3.4a) and PN-RCPN (Fig. 3.4b). We denote, $\mathbf{e}_i^l \in \Re^C$ as the error at enhanced super-pixel nodes; $\mathbf{e}_k^d \in \Re^{2d_s}$ as the error at the decombiner; $\mathbf{e}_k^c \in \Re^{2d_s}$ as the error at the combiner and $\mathbf{e}_i^s \in \Re^{d_s}$ as the error at the semantic mapper. Subscripts *bp* and *total* indicate bypass and the sum total error at a node, respectively. We assume a non-zero categorizer error signal for the first super-pixel only, ie $\mathbf{e}_{i\neq 1}^l = \mathbf{0}$. These assumptions facilitate easier back-propagation tracking through the parse tree, but the conclusions drawn will hold for general cases as well. Under the aforementioned assumptions, we also trace the mathematical relations between various variables, presented in Eq. 3.8, to facilitate the analysis. The variables in red color are indicative of bypass-error paths.

$$\mathbf{e}_6^d = (W_d^T(\mathbf{e}_1^l \bullet f'(\tilde{\mathbf{x}}_1)))_{[1+d_{sem}:end]} \tag{3.8a}$$
$$\mathbf{e}_{1,bp}^s = (W_d^T(\mathbf{e}_1^l \bullet f'(\tilde{\mathbf{x}}_1)))_{[1:d_{sem}]} \tag{3.8b}$$
$$\mathbf{e}_9^d = (W_d^T(\mathbf{e}_6^d \bullet f'(\tilde{\mathbf{x}}_6)))_{[1+d_{sem}:end]} \tag{3.8c}$$
$$\mathbf{e}_{6,bp}^c = (W_d^T(\mathbf{e}_6^d \bullet f'(\tilde{\mathbf{x}}_6)))_{[1:d_{sem}]} \tag{3.8d}$$
$$\mathbf{e}_6^c = (W_c^T(\mathbf{e}_9^d \bullet f'(\mathbf{x}_9)))_{[1:d_{sem}]} \tag{3.8e}$$
$$\mathbf{e}_{6,total}^c = \mathbf{e}_6^c + \mathbf{e}_{6,bp}^c \tag{3.8f}$$
$$\mathbf{e}_1^c = (W_c^T(\mathbf{e}_{6,total}^c \bullet f'(\mathbf{x}_6)))_{[1:d_{sem}]} \tag{3.8g}$$
$$\mathbf{e}_{1,total}^s = \mathbf{e}_{1,bp}^s + \mathbf{e}_1^c \tag{3.8h}$$
$$\mathbf{g}_1^s = (\mathbf{e}_{1,total}^s \bullet f'(\mathbf{x}_1))\mathbf{v}_1^T \tag{3.8i}$$

here, $\mathbf{y}_{m:n}$, is a sub-vector of $\mathbf{y}$ from index m to n, $f'()$ is the derivation of the non-linearity and $a \bullet b$ is the element-wise product.

The first obvious benefit of using pure-nodes is more labeled samples from the same training data that can improve generalization. The second advantage of PN-RCPN can be understood by contrasting the back-propagation signals for a sample image for RCPN and PN-RCPN, with the help of Fig. 3.4a (RCPN) and 3.4b (PN-RCPN). Note that in the case of RCPN, the back-propagated training signal was generated at the enhanced leaf-node features and progressively attenuates as it back-propagates through the parse tree, shown with the help of variable thickness solid red arrows. On the other hand, pure-node RCPN has an internal node (shown as a green color node) that injects a strong error signal deep into the parse tree, resulting in stronger gradients even in the deeper layers. Moreover, PN-RCPN *explicitly* forces the combiner to learn meaningful combination of two super-pixels, because incorrect classification of the combined features is penalized.

Now, we come to the fourth benefit of the PN-RCPN architecture. In what follows, we describe a subtle yet potentially serious problem related to RCPN learning, provide empirical evidence that this problem exists, and argue that PN-RCPN can offer a solution to this problem.

3.6.1.1 Understanding the Bypass Error

During the minimization of the loss functions (Eqn. 3.5 or 3.7), typically, more effective parameters in bringing down the objective function receive stronger gradients and reach their stable state early. Due to the presence of multiple layers of non-linearities and complex connections, the loss function is highly non-convex and the solution inevitably converges to a local minimum. It was shown in [112] that the combiner and decombiner assembly is the most important constituent of the RCPN model. Therefore, we expect the learning process to pay more attention to $\mathbf{W}_c$ and $\mathbf{W}_d$. Unfortunately, the RCPN architecture introduces short-cut paths in the computation graph from the semantic mapper to the categorizer during the forward propagation that gives rise to *bypass errors* during back-propagation. Bypass errors severely affect the learning by reducing the effect of the combiner on the overall loss function, thereby favoring a non-desirable local minimum.

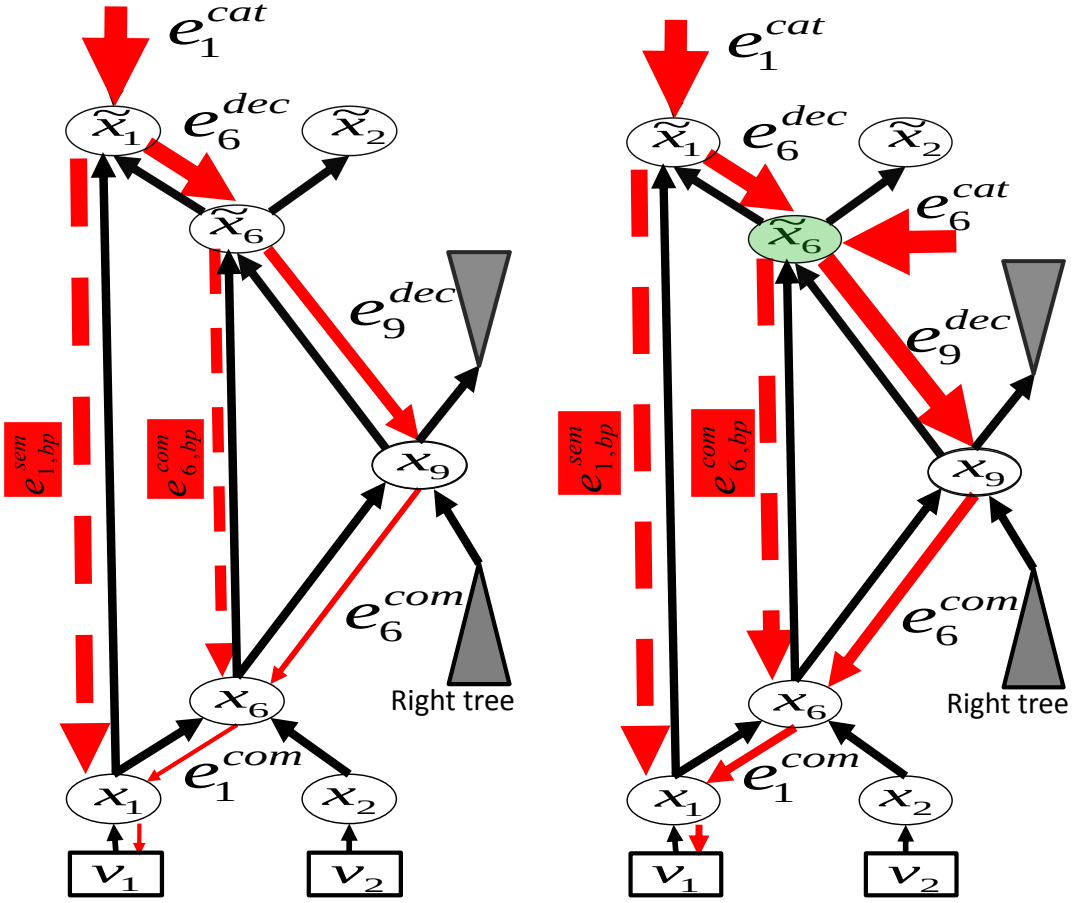
In order to understand the effect of bypass error, we again make use of the example in Fig. 3.4 to show that bypass paths allow the back-propagated error signals from the categorizer ($\mathbf{e}_i^l$) to reach the semantic mapper through one layer only. On the other hand,

$\mathbf{e}_i^l$ goes through multiple layers before reaching the combiner. Therefore, the gradient $\mathbf{g}_c$ for the combiner is weaker than the gradient for the semantic mapper ($\mathbf{g}_s$).

From the Fig. 3.4a we can see that there are two possible paths for $\mathbf{e}_1^c$ to reach the combiner. One of them requires 2 layers ($\tilde{\mathbf{x}_1} \rightarrow \tilde{\mathbf{x}_6} \rightarrow \mathbf{x}_6$) and the other requires 3 layers ($\tilde{\mathbf{x}_1} \rightarrow \tilde{\mathbf{x}_6} \rightarrow \mathbf{x}_9 \rightarrow \mathbf{x}_6$). Similarly, $\mathbf{e}_1^c$ can reach $\mathbf{x}_1$ through a 1 layer bypass path ($\tilde{\mathbf{x}}_1 \rightarrow \mathbf{x}_1$) or a several layers path through the parse tree. Due to gradient attenuation and non-expansive nature of non-linearities, the smaller the number of layers the stronger the back-propagated signal, therefore, bypass errors lead to $\mathbf{g}_s \geq \mathbf{g}_c$. This can potentially render the combiner network inoperative and guide the training towards a network that effectively consists of a $N_{sem} + N_{dec} + N_{cat}$ layer network from the visual feature ($\mathbf{v}_i$) to the super-pixel label ($\mathbf{y}_i$). This results in little or no contextual information exchange between the super-pixels. In the worst case $W_d = [W \;\; 0]$; this removes the effect of parents on their children features during top-down contextual propagation through the decombiner, thereby completely removing the effect of the combiner from RCPN. Practically, the random initialization of the parameters ensures that they will not converge to such a pathological solution. However, we show that a better local minimum can be achieved by tackling the bypass errors.

In order to see that $\mathbf{g}_s \geq \mathbf{g}_c$, we compute the gradient strengths of each module ($\mathbf{g}_s$, $\mathbf{g}_c$, $\mathbf{g}_d$, $\mathbf{g}_l$) during training. The gradient strengths of different modules for RCPN and PN-RCPN are normalized by the number of parameters and plotted in Fig. 3.5a and Fig. 3.5b, respectively. As expected, $\mathbf{g}_l$ is the strongest, because it is closest to the initial error signal. Surprisingly, for RCPN $\mathbf{g}_s$ is slightly stronger than $\mathbf{g}_d$ and significantly stronger than $\mathbf{g}_c$ during the initial phase of training. Normally, we would expect $\mathbf{g}_s$, which is the farthest away from the error signal, to be the weakest due to vanishing gradients. This observation suggests that the initial training phase favors a multi-layer NN. However, we also observe that during the later stages of training, $\mathbf{g}_c$ is comparable to other gradients. Unfortunately, it has been conclusively established, by many empirical studies, that the initial phase of training is crucial for determining the final values of the network parameters, and thereby their performance [26]. From the figure we see that the combiner catches up with the other modules during later stages of training, but by then the parameters are already in the attraction basin of a poor solution.

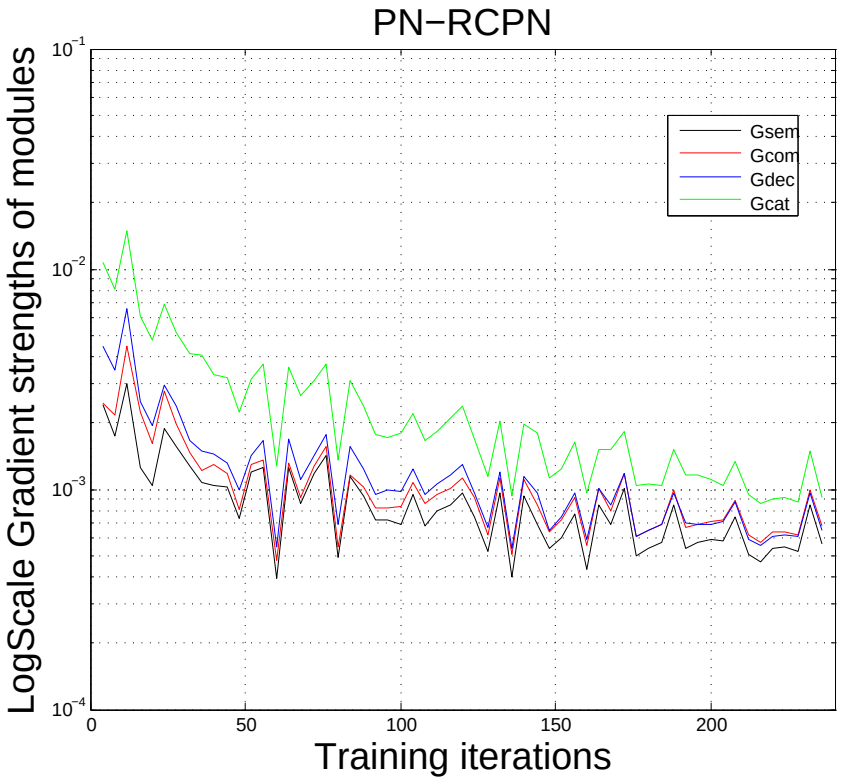
On the other hand, the gradients for PN-RCPN (Fig 3.5b) follow the natural order of strength, which gives more importance to the combiner and decombiner than the semantic mapper during the initial training. Fig. 3.4b provides an intuitive explanation by showing the categorizer error signal ($\mathbf{e}_6^l$) for $\tilde{\mathbf{x}}_6$ that reaches to the combiner through one layer only ($\mathbf{e}_{6,bp}^c$). To further investigate which of the three aforementioned benefits play the biggest role in improving the performance of PN-RCPN over RCPN, we trained PN-RCPN on SIFT flow under the same setting as Table 3.2, but we removed as many leaf node labels from the classification loss as the number of pure-nodes. This makes the number of labeled samples equal in both RCPN and PN-RCPN, but leaf-nodes are replaced with pure-nodes. As expected, it still improves PPA and MCA score for PN-RCPN (80.5% and 35.3%) vs. RCPN (79.6% and 33.6%). This last experiment confirms that inclusion of pure-nodes does not only provide more samples but also helps in overcoming the discussed shortcomings of RCPN.



(a) RCPN (b) PN-RCPN

Figure 3.4: Back-propagated error tracking to visualize the effect of bypass error. The variables follow the notation introduced in Sec. 3.6.1. Forward propagation and back-propagation are shown by solid black and red arrows, respectively. The attenuation of the error signal is shown by variable **width** red arrows. The bypass errors are shown with dashed red arrows. (a) RCPN: Error signal from $\tilde{\mathbf{x}}_1$ reaches to $\mathbf{x}_1$ in just one step, through the bypass path. (b) PN-RCPN introduces pure-nodes classification loss (for $\tilde{\mathbf{x}}_6$), thereby, forcing the network to learn meaningful internal node representation via combiner, thereby, promoting effective contextual propagation.

(a) RCPN gradient strength



(b) PN-RCPN gradient strength

Figure 3.5: Comparison of gradient strengths of different modules of (a) RCPN and (b) PN-RCPN during training.

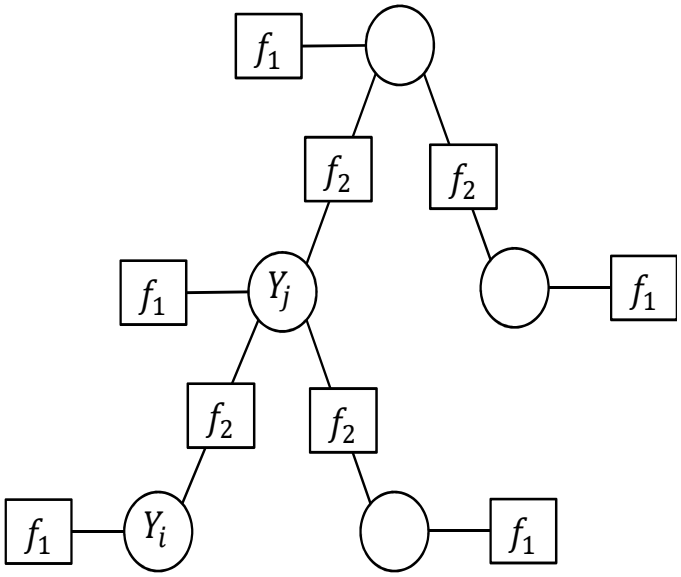


Figure 3.6: Factor graph representation of the MRF model.

3.7 Tree-MRF RCPN

The pure node extension of RCPN provides the label distributions over merged-regions associated with the internal nodes in addition to individual super-pixel labels. In this section, we describe a Markov Random Field (MRF) structure to model the output label dependencies of the super-pixels while leveraging the internal node label distributions for hierarchical consistency. The proposed MRF uses the same trees structure as that of the parse trees used for RCPN inference. A factor graph representation of this MRF is shown in Figure 3.6. The variables Y_i are L-dimensional binary label vectors associated with each region of the image, L is the number of possible labels. The k^{th} dimension of Y_i is set according to the presence (1) or absence (0) of the k^{th} class super-pixel in the region.

The unary potentials f_1 are given by the label distributions predicted by the RCPN and defined as -

$$f_1(Y_i) = \frac{-\mathbf{Y_i}^T \log(\mathbf{p}_i)}{\|Y_i\|_1} \tag{3.9}$$

where $\mathbf{p}_i$ is the softmax output of the categorizer network for super-pixel i. If the probabilities given by RCPN are not degenerate, the unary potential prefers to assign a single label, that of the node with the highest probability.

The pairwise potentials f_2 are introduced to impose consistency between a pair of child and parent regions. The parent region *must* include all the labels assigned to its children regions, which is a hard constraint:

$$f_2(Y_i, Y_j) = \begin{cases} \infty, & \text{if } \mathcal{S}(Y_i) \setminus \mathcal{S}(Y_j) \neq \emptyset. \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

where node j is the parent node of i and $\mathcal{S}(Y)$ is the set of all the labels in the merged-region with label vector Y.

The unary potentials f_1 utilize all levels of the tree simultaneously and prefer purer nodes, whereas pairwise potentials, f_2 enforce consistency across the tree hierarchy. This design allows for spatial smoothness at lower levels and mixed labeling at the higher levels. The tree structure of the MRF affords exact decoding using max-product belief propagation. The size of the state space is exponential in the number of labels. However, in practice there are rarely more than a handful of different object classes within an image. Therefore, to reduce the size of the state space, we first identify different labels predicted by the RCPN and only retain the 9 most frequently occurring super-pixel labels per image.

3.8 Experimental analysis

In this section we evaluate the performance of proposed methods for semantic segmentation on three different datasets: Stanford Background, SIFT Flow and Daimler Urban. Stanford background dataset contains 715 color images of outdoor scenes, it has 8 classes and the images are approximately 240×320 pixels. We used the 572 train and 143 test image split provided by [119] for reporting the results. SIFT Flow contains 2688, 256×256 color images with 33 semantic classes. We experimented with the train/test (2488/200) split provided by the authors of [130]. Daimler Urban dataset has 500, 400×1024 images captured from a moving car in a city, it has 5 semantic classes. We trained the model using 300 images and tested on the rest of the 200 images, the same split-ratio has been used by previous work on this dataset.

3.8.1 Visual feature extraction

We use a Multi-scale convolution neural network (Multi-scale CNN) [28] to extract pixel-wise features using the publicly available library Caffe [52]. We follow [112] and use the same CNN structure with similar preprocessing (subtracting 0.5 from each channel at each pixel location in the RGB color space) at 3 different scales (1,1/2 and 1/4) to obtain the visual features. The CNN architecture has three convolutional stages with $8 \times 8 \times 16$ *conv* $\rightarrow 2 \times 2$ *maxpool* $\rightarrow 7 \times 7 \times 64$ *conv* $\rightarrow 2 \times 2$ *maxpool* $\rightarrow 7 \times 7 \times 256$ *conv* configuration, each max-pooling is non-overlapping. Therefore, every image scale gives a 256 dimensional output map. The outputs from each scale are concatenated to get the final feature map. Note that the $256 \times 3 = 768$ dimensional concatenated output feature map is still 1/4th of the height and width of the input image due to the max-pooling operations. In order to obtain the input size per-pixel feature map we simply scale-up each feature map by a factor of 4 in height and width using bilinear interpolation, .

We use the publicly available implementation of [73] to obtain 100 (same as RCPN) and 800 super-pixels per image for SIFT Flow and Daimler Urban, respectively. Daimler uses more super-pixels due to its larger size. For Stanford background, we have used the super-pixels provided by [119].

3.8.2 Model Selection

Unlike most of the previous works that rely on careful hand-tuning and expert knowledge for setting the model parameters, we only need to set one parameter, namely d_{sem}, after we have fixed the modules to be 1-layer neural networks. This affords a generic

approach to semantic segmentation that can be easily trained on different datasets. For the sake of strict comparison with the original RCPN architecture, we also use 1-layer modules with $d_{sem} = 60$ in all our experiments. *Plain-NN* refers to training a 2-layer NN with 60 hidden nodes, on top of visual features for each super-pixel. *RCPN* refers to the original RCPN model [112]. *PN-RCPN* refers to pure-node RCPN and *TM-RCPN* refers to tree-MRF RCPN.

3.8.3 Evaluation metrics

We have used four standard evaluation metrics -

- **Per pixel accuracy (PPA):** Ratio of the correct pixels to the total pixels in the test images, while ignoring the background.
- **Mean class accuracy (MCA):** Mean of the category-wise pixel accuracy.
- **Intersection over Union (IoU):** Ratio of true positives to the sum of true positive, false positive and false negative, averaged over all classes. This is a popular measure for semantic segmentation of objects because it penalizes both over- and under-segmentation.
- **Time per image (TPI):** Time required to label an image on GPU and CPU.

The results from previous works are taken directly from the published articles. Some of the previous works do not report all four evaluation metrics; we leave the corresponding entry blank in the comparison tables.

3.8.4 Stanford Background

We report our results with CNN features extracted from the original scale only, because multi-scale CNN features overfit, perhaps due to small training data, as observed in [112]. We use 10 and 40 random trees for training and testing, respectively. The results are shown in Table 3.1. From the comparison, it is clear that our proposed approaches outperform previous methods. We observe that PN-RCPN significantly improves the results in terms of MCA and IoU over RCPN. We observe a marginal improvement offered by TM-RCPN over PN-RCPN.

3.8.5 SIFT Flow

We report our results using multi-scale CNN features at three scales (1,1/2 and 1/4), as in [112]. Some of the classes in the SIFT Flow dataset have a very small number of training instances, therefore, we also trained with balanced sampling to compensate for rare occurrence, referred to as *bal.* prefix. We use 4 and 20 random trees for training and testing, respectively. The results for SIFT flow dataset are shown in Table 3.2. PN-RCPN led to significant improvement in all three measures over RCPN and balanced training led to a significant boost in MCA. The use of TM-RCPN does not affect the results much compared to PN-RCPN. We observe a strong trade-off between PPA and MCA on this dataset. Our overall best model in terms of both PPA and MCA (*bal. TM-RCPN*) looks equivalent to the work in [139]; PPA: 76.4 vs. 79.8, MCA: 52.6 vs. 48.8.

Table 3.1: Stanford background result.

Method	**PPA**	**MCA**	**IoU**	**TPI (s)** CPU/GPU
Gould, [34]	76.4	NA	NA	30 – 600 / NA
Munoz, [83]	76.9	NA	NA	12 / NA
Tighe, [130]	77.5	NA	NA	4 / NA
Kumar, [59]	79.4	NA	NA	$\leq$ 600 / NA
Socher, [119]	78.1	NA	NA	NA / NA
Lempitzky, [63]	81.9	72.4	NA	$\geq$ 60 / NA
Singh, [118]	74.1	62.2	NA	20 / NA
Farabet, [28]	81.4	76.0	NA	60.5 / NA
Eigen, [31]	75.3	66.5	NA	16.6 / NA
Pinheiro, [87]	80.2	69.9	NA	10 / NA
Plain-NN	80.1	69.7	56.4	1.1/0.4
RCPN [112]	81.8	73.9	61.3	1.1/0.4
PN-RCPN	82.1	79.0	64.0	1.1/0.4
TM-RCPN	**82.3**	**79.1**	**64.5**	1.6–6.1/0.9–5.9

3.8.6 Daimler Urban

We report our results using multi-scale CNN features with balanced training. We would like to emphasize that previously reported results make use of depth information and/or visual odometry and yet we outperform them significantly. For this dataset previous works have not reported PPA and MCA, therefore, we drop PPA and report IoU for all five classes (IoU) and for dynamic objects ie cars and pedestrians (IoU Dyn). The results for Daimler Urban dataset are shown in Table 3.3. Simply using the multi-scale CNN super-pixel features with a 2-layer NN classifier already outperforms the previous state-of-the-art results. RCPN provides large improvements over the *Plain-NN* and our PN-RCPN improves it further. We observe significant improvements in terms of IoU with the use of PN-RCPN over RCPN and Plain-NN. We believe that the reason for such a dramatic improvement is the well structured image semantics of the dataset that allows RCPN and PN-RCPN to learn the structure very effectively and utilize the context in a much better way than the other two datasets. Some of the representative segmentation results are shown in Fig. 5. We have also submitted a complete video of semantic segmentation for all the test images for Daimler urban in the supplementary material.

3.8.7 Segmentation Time

In this section we provide the timing details for the experiments. Due to similar image sizes, SIFT flow and Stanford Background took almost the same computation per image except while using TM-RCPN, because of the difference in label state-space size. The time break-up for SIFT flow (same for Stanford) in seconds is 0.3 (super-pixellation) + 0.08/0.8 (GPU/CPU visual feature) + 0.01 (PN-RCPN) + 0.5–5 (TM-MRF). For Daimler, the corresponding timings are 2.4 + 0.4/3.5 + 0.09 + 6 seconds. Therefore, the bottleneck for our system is the super-pixellation time for PN-RCPN and MRF inference for TM-RCPN. Fortunately, there are real-time super-pixellation algorithms, such as [30], that can help us achieve state-of-the-art semantic segmentation within 100 milliseconds on

Table 3.2: SIFT Flow result. The last row shows the results of a very deep CNN network based semantic segmentation approach that was published during the preparation of this dissertation.

Method	**PPA**	**MCA**	**IoU**	**TPI (s)** CPU/GPU
Tighe, [130]	77.0	30.1	NA	8.4 / NA
Liu, [71]	76.7	NA	NA	31 / NA
Singh, [118]	79.2	33.8	NA	20 / NA
Eigen, [31]	77.1	32.5	NA	16.6 / NA
Farabet, [28]	78.5	29.6	NA	NA / NA
(Balanced), [28]	72.3	50.8	NA	NA / NA
Tighe, [129]	78.6	39.2	NA	$\geq$ 8.4 / NA
Pinheiro, [87]	77.7	29.8	NA	NA / NA
Yang, [139]	79.8	48.7	NA	$\leq$ 12/NA
Plain-NN	76.3	32.1	24.7	1.1/0.36
RCPN, [112]	79.6	33.6	26.9	1.1/0.4
bal. RCPN, [112]	75.5	48.0	28.6	1.1/0.4
PN-RCPN	**80.9**	39.1	30.8	1.1/0.4
bal. PN-RCPN	75.5	**52.8**	30.2	1.1/0.4
TM-RCPN	80.8	38.4	30.7	1.6–6.1/0.9–5.4
bal. TM-RCPN	76.4	52.6	**31.4**	1.6–6.1/0.9–5.8
FCN-16s [76]	***85.2***	*51.7*	***39.5***	NA/0.2

an NVIDIA Titan Black GPU. We are significantly faster than all the other competing approaches except Stixmantics, which we outperform by a margin of 19%.

3.9 Related Work

We have already provided a brief overview of popular approaches to scene labeling in Sec. 2.1.1. This section discusses the approaches that are more closely related to our approach and brings out the differences and advantages of our approach over previous art. Scene labeling has two broad categories of approaches - non-parametric and model-based. Recently, many non-parametric approaches for natural scene parsing have been proposed [130, 71, 118, 31, 129]. The underlying theme is to find similar looking images to the query image from a database of pixel-wise labeled images, followed by super-pixel label transfer from the retrieved images to the query image. Finally, a structured prediction model such as CRF is used to integrate contextual information to obtain the final labeling. These approaches mainly differ in the retrieval of candidate images or super-pixels, transfer of label from the retrieved candidates to the query image, and the form of the structured prediction model used for final labeling. They are based on nearest-neighbor retrieval that introduces a performance/accuracy trade-off. The variations present in natural scene images are large and it is very difficult to cover this entire space of variation with a reasonable size database, which limits the accuracy of these methods. On the other extreme, a very large database would require large retrieval-time, which limits the scalability of these methods.

Model-based approaches learn the appearance of semantic categories and relations

Table 3.3: Daimler result.

Method	**MCA**	**IoU**	**IoU Dyn**	**TPI (s)** CPU/GPU
Joint-ALE, [61]	NA	72.6	63.7	NA / 111
Depth-ICF, [32]	NA	52.8	44.2	NA/3.2
SLICbaseline, [102]	NA	50.2	45.1	NA/0.5
StixBaseline, [102]	NA	63.9	59.8	NA/0.5
Stixmantics, [101]	NA	66.9	62.6	NA/0.05
bal. Plain-NN	85.1	78.5	60.6	5.9 / 2.8
bal. RCPN	89.3	83.1	69.4	6.0 / 2.8
bal. PN-RCPN	**91.2**	**85.9**	**75.9**	6.0 / 2.8
bal. TM-RCPN	91.0	**85.9**	75.7	12 / 8.8

among them using a parametric model. In [34, 83, 81, 80], CRF models are used to combine unary potentials devised through the visual features extracted from super-pixels with the neighborhood constraints. The differences are mainly in terms of the visual features, unary potentials and the structure of the CRF model. Lempitsky et al. [63] have formulated a joint-CRF on multiple levels of an image segmentation hierarchy to achieve better results than a flat-CRF on the image super-pixels only. The CRF based models are among the most popular and accurate semantic segmentation approaches, but they all suffer from the limitation of inference algorithms and the forms of potential functions, please refer to Sec. 3.1. In contrast, in our model, we can efficiently learn complex relations between a single node label and all the observations from an image, allowing a large context to be considered effectively. Additionally, the inference procedure is a simple feed-forward pass that can be performed very fast. However, the form of our function is still a unary term and our model cannot represent higher order label dependencies. Our model can also be used to obtain the unary potential for a structured inference model.

Socher et al. [119] learned a mapping from the visual features to a semantic space followed by a two-layer neural network for classification. Better use of contextual information, with the same super-segments and features, increased the performance on Stanford background dataset from the CRF based method of Gould et al. to semantic mapping of Socher et al. to the proposed work (76.4% $\rightarrow$ 78.1% $\rightarrow$ 81.4%). It indicates that the neural network based models have the potential to learn more complicated interactions than a CRF. Moreover, NN based approaches have the advantage of being extremely fast, due to the feed-forward nature. Farabet et al. [28] proposed to learn the visual features from raw-image/label training pairs using a multi-scale convolutional neural network (Multi-CNN). They reported state-of-the-art results on various datasets using gPb, purity-cover and CRF on top of their learned features. Pinheiro et al. [87] extended their work by feeding in the per-pixel predicted labels using a CNN classifier to the next stage of the same CNN classifier. However, their propagation structure is not adaptive to the image content and only propagating label information did not improve much over the prior work. Similar to these methods, we also make use of the trainable Multi-CNN module to extract local features in our pipeline. However, our novel context propagation network shows that propagating semantic representation bottom up and top down using a parse three hierarchy is a more effective way to aggregate global context information. Please see Tables 3.2 and 3.1 for a detailed comparison of our method with the methods discussed above.

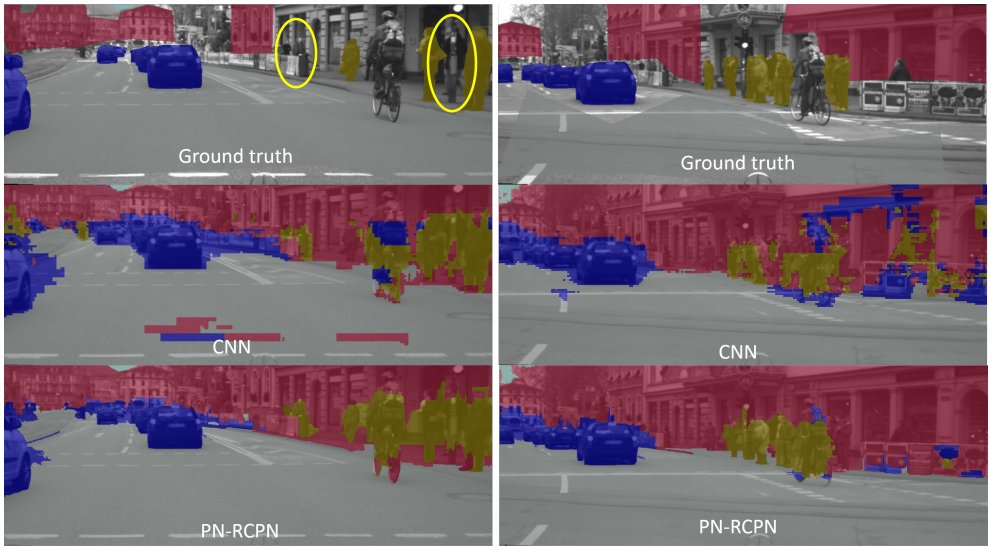


Figure 3.7: Some representative image segmentation results on Daimler Urban dataset. Here, CNN refers to direct per-pixel classification resulting from the multi-scale CNN. The images are only partially labeled and we have shown the unlabeled pedestrians by yellow ellipses.

In parallel with our work, a very deep neural network based semantic segmentation approaches has been proposed in [76], referred to as FCN-16s. It makes use of the pre-trained CNN filters of a 19-layer network for object classification [117]. FCN-16s computes pixel-wise sum of the feature maps after the 4^{th} and 5^{th} stages of 2×2 max-pooling operations, after appropriate resizing using learned interpolation kernels, and feeds the summed features to a semantic classifier. The pre-trained CNN layers are fine-tuned for semantic segmentation loss like the Multi-CNN training to obtain the final segmentation system. It is also an end-to-end trainable system, like us. FCN-16s has reported much better performance than our approach on SIFT Flow dataset, please refer to Table 3.2. We believe that it is due to the large-size FOV for each pixel (156×156) and highly non-linear pre-trained filters. Unfortunately, the pre-training stage requires a large amount of training data in similar domain, 14 Million labeled images for this case, that may prohibit the feasibility of this approach where training data is limited. Even with this serious limitation, it makes a strong case to use pre-trained visual features with our approach to assess the advantages of the proposed RCPN structure for context utilization.

3.10 Conclusion

We introduced a novel deep neural network architecture, which is a combination of a convolutional neural network and recursive neural network, for pixel-wise semantic scene labeling. The key contribution is the recursive context propagation network, which effectively propagates contextual information from one location of the image to other locations in a feed-forward manner. We further analyzed the recursive contextual propagation network and discovered potential problems with the learning of its parameters. Specifically, we showed the existence of bypass errors and explained how it can reduce the RCPN model to an effective multi-layer neural network for each super-pixel. Based on our findings, we

proposed to include the classification loss of pure-nodes to the original RCPN formulation and demonstrated its benefits in terms of avoiding the bypass errors. We also proposed a tree MRF on the parse tree nodes to utilize the pure-node's label estimation for inferring the super-pixel labels. The proposed approaches lead to impressive performance on three segmentation datasets: Stanford background, SIFT flow and Daimler urban.

Chapter 4

Multi-modal face recognition using PLS to learn the common representation

We present a novel way to perform multi-modal face recognition by using Partial Least Squares (PLS) to linearly map images in different modalities to a common linear subspace in which they are mapped to nearby locations. PLS has been previously used effectively for feature selection in face recognition. We show both theoretically and experimentally that PLS can be used effectively across modalities. We also formulate a generic intermediate subspace comparison framework for multi-modal recognition. Surprisingly, we achieve high performance using only pixel intensities as features. We experimentally demonstrate the highest published recognition rates on the pose variations in the PIE data set, and also show that PLS can be used to compare sketches to photos, and to compare images taken at different resolutions.

4.1 Motivation

In face recognition, one often seeks to compare gallery images taken under one set of conditions, to a probe image acquired differently. For example, in criminal investigations, we might need to compare mug-shots to a forensic sketch drawn by a sketch artist based on the verbal description of the suspect. Similarly, mug-shots or passport photos might be compared to surveillance images taken from a different viewpoint. The probe image might also be of lower resolution (LR) compared to a gallery of high resolution (HR) images. All these situations are simply different instances of cross-modal matching and call for a common representation framework.

4.2 Related Work

There has been a huge amount of prior work on comparing images taken in different modalities, which we can only sample here. In much of this work, images taken in one modality are automatically converted to the second modality prior to comparison. For example a holistic mapping [127] is used to convert a photo image into a corresponding sketch image. In [137, 135, 74] the authors have used local patch based mappings to convert images from one modality to the other for sketch-photo recognition. Since the mapping from one modality to the other is generally non-linear, local patch based approaches generally perform better than the global ones because they can approximate the non-linearity in a better manner. Producing good quality high-resolution (HR) face images from very low-resolution (LR) noisy surveillance videos is an important area of study owing to its importance for security reasons. The work in [140] proposed a holistic approach for hallucinating HR face images from LR images. Some local patch-wise based approaches were also proposed in [70, 67, 142] to hallucinate a HR face image from a given LR face image. The comparison between the holistic and local approaches reveals that local approaches perform better. For face recognition with pose and lighting variation [36, 19, 97], 3D knowledge of faces is used to warp an off-axis image to a frontal image, and

to normalize lighting prior to comparison. These approaches may use representations that are specific to a domain, or may employ a more general, learning-based approach, that typically requires corresponding patches in the training set [36, 19, 96, 97]. Our approach does not attempt to synthesize images of one modality from another. While excellent work has been done on synthesis, this may in principle be an ill-posed problem that is more difficult than simply comparing images taken in two different modalities. A second approach is to compare images using a representation that is insensitive to changes in modality. For example, [57] used SIFT feature descriptors and multi-scale local binary patterns to represent image and sketch of faces then performed recognition based on this common representation. This approach worked well because both SIFT and LBP features extract gradient information that is approximately the same in both photo and sketch at corresponding positions. While some descriptors, such as SIFT, are robust across a range of variations in modalities, no single representation can be expected to handle all variations in modality. Two prior methods are closer to our work in spirit, and have provided valuable inspiration. In [128] the authors have used Singular Value Decomposition to derive a common content space for a set of different styles and [12] uses a probabilistic model to generate coupled subspaces for different poses. Recently, [64] used CCA to project images in different poses to a common subspace and compared them using probabilistic modeling. While related our approach is different in several ways: we achieve strong results using simple pixel intensities, without probabilistic modeling of patches; we show theoretically why projection methods can handle pose variation; and we show that PLS can outperform CCA with pose variation.

4.3 Proposed Approach

We propose a general framework based on the common representation hypothesis that uses Partial Least Squares (PLS) [99, 98, 1] to perform recognition in a wide range of multi-modal scenarios. PLS has been used before for face recognition, but in a different manner, with different motivation [23, 7, 122, 69, 105]; our contribution is to show how and why PLS can be used for cross-modal recognition. More generally, we argue for the applicability of linear projection to a common subspace for multi-modal recognition. One consequence of our approach is that we do not need to synthesize an artificial gallery image from the probe image. Experimental evaluation of our framework using PLS with pose variation has shown significant improvements in terms of accuracy and run-time over the state-of-art on the CMU PIE face data set [116]. For sketch-photo recognition, our method is comparable to the state of-art. We also illustrate the potential of our method to handle variation in resolution with a simple, synthetic example. In all three domains we apply exactly the same algorithm, and use the same, simple representation of images. Our generic approach performs either similar or better than state-of-the-art approaches that have been designed for specific cross-modal conditions.

Our approach matches probe and gallery images by linearly projecting them into a common space where images with the same identity map to nearby locations, see Fig 4.1. We argue that for a variety of cross-modal recognition and matching problems, such projections will exist and can be found using any of the discussed techniques such as: PLS, CCA, BLM and TFA. In order to decide the optimal technique to obtain the common subspace we theoretically compare the objective functions of PLS, CCA and BLM to bring out the differences.

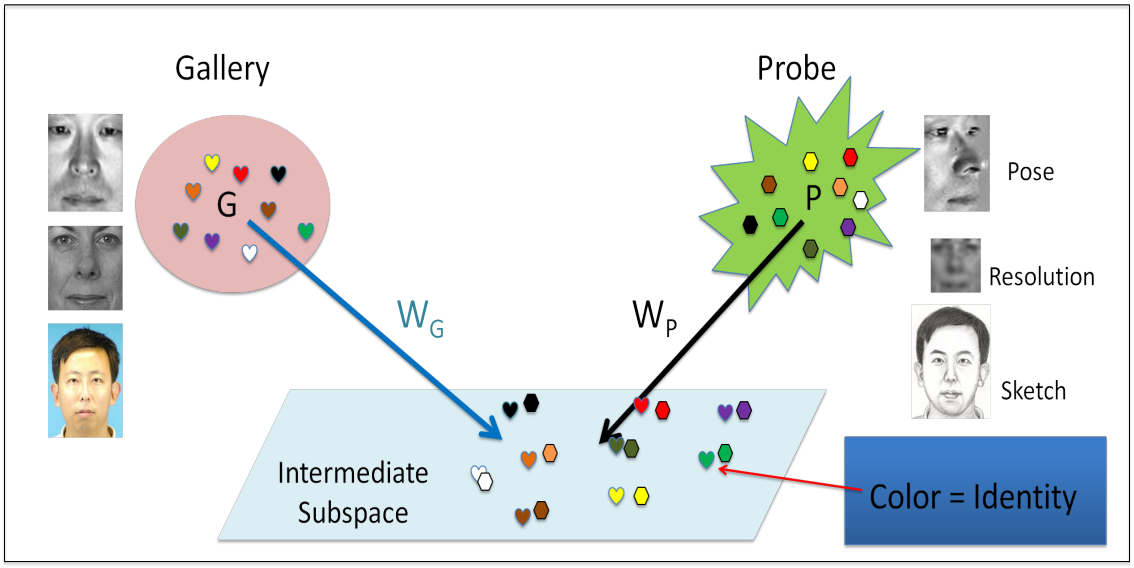


Figure 4.1: Common representation framework for multi-modal face recognition, W_g and W_p are learned using some learning method with training images in gallery and probe modalities.

4.3.1 When can the common representation hypothesis work?

We will use PLS to find projection directions w_p and w_q that map images taken in modality p and q into a common subspace. PLS will seek w_p and w_q that tend to produce high levels of covariance in the projection of corresponding images from different modalities. However, PLS cannot be expected to lead to effective recognition when such projections do not exist. In this section, we show some conditions in which projections of images from two modalities exist in which the projected images are perfectly correlated (and in fact equal). Then we show that these conditions hold for some interesting examples of cross-modality recognition. We should note that the existence of such projections is not sufficient to guarantee good recognition performance. We will assess the actual performance of PLS empirically, in the next section.

4.3.1.1 Existence of correlated projections

In a number of cases, images taken in two different modes can be viewed as different, linear transformations of a single canonical object. Let I_p^k and I_q^k denote column vectors containing the pixels of corresponding images in modalities p and q, respectively. We denote by R^k a matrix (or column vector) that contains a canonical version of I_p^k and I_q^k, such that we can write:

$$\begin{aligned} I_p^k &= AR^k \\ I_q^k &= BR^k \end{aligned} \tag{4.1}$$

for some matrices A and B. We would like to know when it will be possible to find vectors w_p and w_q that project sets of images into a 1D space in which they are highly correlated. We consider a simpler case, looking at when the projections can be made equal. That is,

when we can find w_p and w_q such that for any I_p^k and I_q^k satisfying Eqn 4.1 we have:

$$w_p^T I_p^k = w_q^T I_q^k \quad \Rightarrow \quad w_p^T A R_k = w_p^T B R_k \tag{4.2}$$

$$\Rightarrow w_p^T A = w_p^T B \tag{4.3}$$

Eqn 4.2 can be satisfied if and only if the row spaces of A and B intersect, as the LHS of the Eqn 4.3 is a linear combination of the rows of A, while the RHS is a linear combination of the rows of B. We now give some examples in which this condition holds.

4.3.1.2 High resolution vs. low resolution

For this situation, we can assume that the ideal image is just the high resolution image, so that A is simply the identity matrix, and $I_p^k = R^k$. I_q^k then, can be obtained by smoothing R^k with a Gaussian filter, and sub-sampling the result. Both operations can be represented in matrix form. Any convolution can be represented as a matrix multiplication. For this, the i^{th} row of B contains a vectorized Gaussian filter centered at the image location of the i^{th} pixel in R^k. B can sub-sample the result of this convolution by simply omitting rows corresponding to pixels that are not sampled. Now because A is the identity matrix, it has full rank, and its row space must intersect that of B.

4.3.1.3 Pose variation

We now consider the more challenging problem that arises when comparing two images taken of the same 3D scene from different viewpoints. This raises problems of finding a correspondence between pixels in the two images, as well as accounting for occlusion. To work our way up to this problem, we first consider the case in which there exists a one-to-one correspondence between pixels in the image, with no occlusion.

Permutations: In this case, we can again suppose that A is the identity matrix. In this case, B will be a permutation matrix, which changes the location of pixels without altering their intensities. In this case, A and B are both of full rank, and in fact have a common row space. So again, there exist w_p and w_q that will project I_p^k and I_q^k into a space where they are equal.

Stereo: We now consider a more general problem that is commonly solved by stereo matching. Suppose we represent a 3D object with a triangular mesh. Let R^k contain the intensities on all faces of the mesh that appear in either image (We will assume that each pixel contains the intensity from a single triangle. More realistic rendering models could be handled with slightly more complicated reasoning). Then, to generate images appropriately, A and B will be matrices in which each row contains one 1 and is 0 otherwise. A (or B) may contain identical rows, if the same triangle projects to multiple pixels. The rank of A will be equal to the number of triangles that create intensities in I, and similarly for B. The number of columns in both matrices will equal the number of triangles that appear in either image. So their row spaces will intersect, provided that the sum of their ranks is greater than or equal to the length of R^k, which occurs whenever the images contain projections of any common pixels. As a toy example, we consider a small 1D stereo pair showing a dot in front of a planar background. We might have $I_p^k = [7\ 8\ 2\ 5]$

and $J_q^k = [7\ 2\ 3\ 5]$. In this example we might have $R^k = [7\ 8\ 2\ 3\ 5]$ and:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.4}$$

It can be inferred from the example that row spaces of A and B intersect hence we expect PLS to work. More generally, whenever we are matching two 2-D projections of the same 3-D object, we can think of each image as a linear transformation of the ideal, 3-D object. Therefore, when there is sufficient overlap in the portions of the object that are visible, projection to a common latent space will amount to finding a correspondence between the mutually visible portions of the two images.

4.3.1.4 Comparing images to sketches

Finally, we note that our conditions may approximately hold in the relationship between images and sketches. This is because sketches often capture the edges, or high frequency components of an image. A filter such as a Laplacian of a Gaussian produces an output that is similar to a sketch. Again, the ideal image can be the same as the intensity image, while the sketch image can be produced by a B that represents this convolution, satisfying our conditions.

4.3.2 Difference between PLS, BLM and CCA

BLM, CCA and PLS try to achieve the same goal but the difference in their objective functions leads to different properties. BLM tries to preserve the variance present in different feature spaces and does not explicitly try to make projected samples similar. It is interesting to compare the objective function of PLS with that of CCA to emphasize the difference between the two. CCA tries to maximize the correlation between the latent scores

$$\begin{aligned} [\mathbf{w}_p, \mathbf{w}_q] = &\underset{\mathbf{w}_p, \mathbf{w}_q}{\operatorname{argmax}} \left(corr[Y_p^T \mathbf{w}_p, Y_q^T \mathbf{w}_q]^2\right) \\ &s.t. \quad \|\mathbf{w}_p\| = \|\mathbf{w}_q\| = 1 \end{aligned} \tag{4.5}$$

where,

$$corr(\mathbf{a}, \mathbf{b}) = \frac{cov(\mathbf{a}, \mathbf{b})}{var(\mathbf{a}) var(\mathbf{b})} \tag{4.6}$$

putting the expression from 4.6 into 2.6 we get the PLS objective function as:

$$\begin{aligned} [\mathbf{w}_p, \mathbf{w}_q] = &\underset{\mathbf{w}_p, \mathbf{w}_q}{\operatorname{argmax}} \left([var(Y_p^T \mathbf{w}_p)][corr(Y_p^T \mathbf{w}_p, Y_q^T \mathbf{w}_q)]^2 [var(Y_q^T \mathbf{w}_q)]\right) \\ &s.t. \quad \|\mathbf{w}_p\| = \|\mathbf{w}_q\| = 1 \end{aligned} \tag{4.7}$$

It is clear from (4.7) that PLS tries to correlate the latent score of regressor and response as well as captures the variations present in the regressor and response space too. CCA only tries to correlate the latent score hence CCA may fail to generalize well to unseen testing points and even fails to differentiate between training samples in the latent space under some restrictive conditions. Let's consider a simplified case where PLS will

succeed and both BLM and CCA will fail to obtain meaningful directions - Suppose we have two sets of 3D points X and Y and x_i^j and y_i^j denote the j^{th} element of the i^{th} data point in X and Y. Suppose that the first coordinates of x_i and y_i are pairwisely equal and the variance of the first coordinate is very small and insufficient for differentiating different samples. The second coordinates are correlated with a correlation-coefficient $\rho \leq 1$ and the variance present in the second coordinate is ψ. The third coordinate is almost uncorrelated and the variance is $\gg \psi$.

$$\forall i, x_i^1 = y_i^1 = k \quad \Rightarrow \quad var(X^1) = var(Y^1) = \alpha \ll \psi \tag{4.8}$$

$$corr(X^2, Y^2) = \rho \quad and \quad var(X^2), var(Y^2) \approx \psi \tag{4.9}$$

$$corr(X^3, Y^3) \approx 0 \quad and \quad var(X^3), var(Y^3) \gg \psi \tag{4.10}$$

Under this situation CCA will give the first coordinate as the principal direction which projects all the data points in sets X and Y to a common single point in the latent space, rendering recognition impossible. BLM will find a direction which is parallel to the third coordinate, which preserves the inter-set variance but loses all the correspondence. PLS, however, will opt for the second coordinate, which preserves variance (discrimination) as well as maintains correspondence which is crucial for our task of cross-modal matching.

4.4 Experimental results

In this section we carry out several experiments to compare our PLS based multi-modal face recognition with existing approaches for pose-invariant face recognition, sketch-face recognition and high-low resolution face recognition.

4.4.1 Pose-invariant face recognition

The PLS based framework is used for pose invariant face recognition on CMU PIE dataset which has been used by many researchers previously for evaluation. This database contains 68 subjects in 13 different poses and 23 different illumination conditions. We took subject IDs from 1 to 34 for training and the remaining (35 to 68) for testing. As we are dealing with pose variation only, we took all the images in frontal illumination which is illumination number 12. As a preprocessing step, 4 fiducial points (both eye's centers, nose tip and mouth) were manually annotated and an affine transformation was used to register the faces based on the fiducial points. After all the faces are aligned in corresponding poses we cropped 48×40 facial region. Images were turned into gray-scale and intensity values mapped between 0 to 1 were used as features. The number of PLS factors was set to be 30. Choosing more than 30 did not improve the performance but choosing less than 30 worsens the performance. The resulting approach is termed as PLS^{30}, indicating 30 PLS factors were used. The accuracy for all possible gallery-probe pairs is given in Table 4.1. For comparing our approach with other published works we calculated the average of all gallery-probe pairs and the resulting accuracy is listed in Table 4.2. Some authors have reported their results on CMU PIE data with only frontal pose as gallery and a subset of non-frontal poses as probe. For comparison we also list the gallery and probe setting in Table 4.2. Ridge+(Intensity/Gabor) refers to the approach of [65] with raw intensity and Gabor filter response (with probabilistic local score fusion) at fiducial locations as feature, respectively. Similarly, PLS-(Holistic/Gabors) refers to the use of PLS to learn coupled latent space with raw intensity feature from the whole face and probabilistic fusion of

Table 4.1: CMU PIE accuracy using 1-NN matching and PLS with 30 factors, overall accuracy is **90.08**

Probe→ Gallery↓	c34 -60,0	c31 -45,15	c14 -45,0	c11 -30,0	c29 -15,0	c09 0,15	c27 0,0	c07 0,0	c05 15,0	c37 30,0	c25 45,0	c02 45,15	c22 60,0	**Avg**
c34	-/-	88.0	94.0	94.0	91.0	88.0	91.0	97.0	85.0	88.0	70.0	85.0	61.0	**86.2**
c31	85.0	-/-	100.0	100.0	100.0	88.0	85.0	91.0	85.0	88.0	76.0	85.0	76.0	**88.4**
c14	97.0	100.0	-/-	100.0	97.0	91.0	97.0	100.0	91.0	100.0	82.0	91.0	67.0	**92.8**
c11	79.0	97.0	100.0	-/-	100.0	88.0	100.0	100.0	97.0	97.0	85.0	88.0	67.0	**91.6**
c29	76.0	94.0	100.0	100.0	-/-	100.0	100.0	100.0	100.0	100.0	85.0	91.0	73.0	**93.3**
c09	76.0	88.0	91.0	94.0	94.0	-/-	97.0	94.0	91.0	88.0	82.0	79.0	70.0	**87.2**
c27	85.0	91.0	97.0	100.0	100.0	100.0	-/-	100.0	100.0	100.0	85.0	88.0	79.0	**93.9**
c07	79.0	91.0	97.0	100.0	100.0	97.0	100.0	-/-	100.0	97.0	85.0	91.0	76.0	**92.9**
c05	79.0	97.0	97.0	94.0	100.0	94.0	100.0	100.0	-/-	97.0	91.0	91.0	82.0	**93.6**
c37	79.0	94.0	100.0	94.0	94.0	88.0	94.0	94.0	97.0	-/-	100.0	100.0	94.0	**94.1**
c25	67.0	82.0	76.0	79.0	88.0	88.0	88.0	91.0	94.0	97.0	-/-	97.0	76.0	**85.5**
c02	76.0	88.0	88.0	94.0	94.0	88.0	97.0	94.0	100.0	100.0	100.0	-/-	97.0	**93.1**
c22	64.0	70.0	64.0	79.0	76.0	67.0	82.0	82.0	85.0	91.0	85.0	91.0	-/-	**78.4**

Table 4.2: comparison of PLS with other published work on CMU PIE.

Method	Gallery/Probe	Accuracy	PLS^{30}
Eigenface [35]	all/all	16.6	**90.1**
ELF [35]	all/all	66.3	**90.1**
Bilnear Model	all/all	79.6	**90.1**
CCA	all/all	87.4	**90.1**
FaceIt [35]	all/all	24.3	**90.1**
4ptSMD [16]	all/all	86.8	**90.1**
SlantSMD [17]	all/all	**90.1**	**90.1**
Ridge+Intensity [65]	c27/rest all	88.24	**93.9**
PLS-Holistic [66]	c27/rest all	81.44	**93.9**
Yamada [53]	c27/rest all	85.6	**93.9**
LLR [18]	c27/c(05,07,09,11,37,29)	94.6	**100**
PGFR [75]	c27/c(05,37,25,22,29,11,14,34)	86	**93.4**
Ridge+Gabor [65]	*c27/rest all*	*90.9*	***93.9***
PLS-Gabor [66]	*c27/rest all*	*89.05*	***93.9***
3DMM+LGBP* [6]	*c27/c(11,29,07,09,05,37)*	*99.0*	***100.0***

local scores based on Gabor filter response at fiducial locations, respectively. A simple comparison clearly reveals that PLS^{30} approach outperforms all the methods. It should be noted that the comparison with 3DMM+LGBP [6] is not fair because the results in [6] are reported on 67 subject gallery whereas, we report on 34 subject gallery. However, we still include it for the sake of completeness.

4.4.2 Low resolution vs High resolution

This problem is yet another multi-modal problem because probe images from a surveillance camera are generally low resolution (LR) with slight motion blur and noise. The gallery generally contains high resolution (HR) faces. To verify the applicability of our method we have synthetically generated low resolution images for frontal face images in a subset of FERET face dataset and performed recognition. The original HR images were chosen to be 7666 and different size LR images were tested for recognition. Fig 4.2 shows the recognition accuracy of the proposed method. Note that a direct comparison of HR and LR face images with as low a resolution as 54 resulted in 60% recognition accuracy. Moreover, the number of PLS bases used in any case for optimal performance

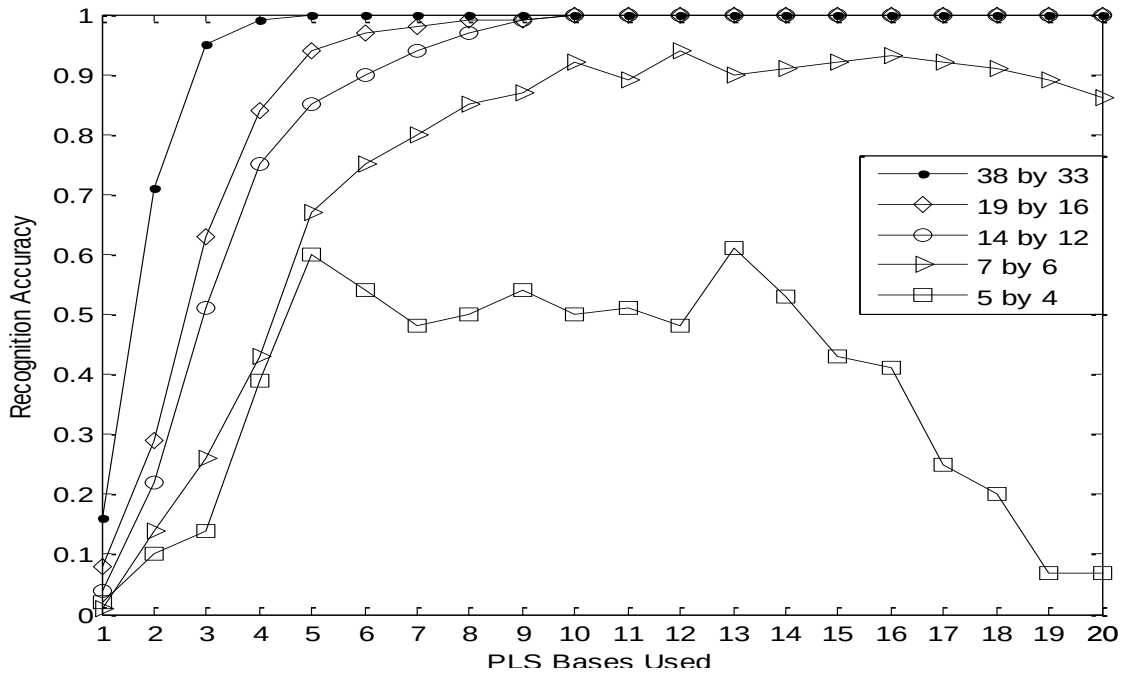


Figure 4.2: Accuracy for Low Resolution face recognition vs. the number of PLS bases used with different size LR images used.

are not greater than 20 and for some cases just 3 PLS bases gave 95% accuracy. We have used 90 faces for training and 100 for testing. Due to lack of space we have not shown the results for BLM but it should be noted that it performed similarly. However, performance of CCA was very poor ranging between 30-50% only.

4.4.3 Sketch-face recognition

To demonstrate the generality of our proposed approach we have also tested it on a sketch vs. photo recognition problem. To test the performance of our method we have used a subset of the CUHK sketch - face dataset [3]. We used a subset containing 188 subjects' face images and corresponding hand drawn sketch pairs. 88 sketch-photo pairs were used as the training sample and the remaining 100 were used as the testing set. We formed 5 random partitions of the dataset to generate different sets of training and testing data and report the average accuracy. In this case, we have used 70 PLS bases and 50 eigenvectors for the Bilinear Model. A comparison of our method with other reported results is shown in Table 4.

From the comparison it is clear that in spite of being holistic in nature, the proposed method achieves respectable accuracy. We feel that this is encouraging because our method is completely general; we have used exactly the same algorithm for pose, LR face recognition and sketch. The table also reflects the trend that accuracy is increasing continuously as we move down from holistic to pixel level representation. So it may be possible that using patch-wise features with our method will improve the accuracy. It should be noted that in [5] and [4] the authors have used strong classifiers after extracting patch-wise and pixel based features, whereas we have simply used the NN metric after latent score extraction.

Table 4.3: Sketch-Photo pair recognition accuracy.

Method	**Testing set**	**Type**	**Accuracy**
Wang	100	Holistic	81
Liu	300	Path-wise	87.7
Klare	300	Pixel-wise	99.5
PLS	100	Holistic	93.6
Bilinear	100	Holistic	94.2
CCA	100	Holistic	94.6

4.5 Conclusion

We have demonstrated a general common subspace framework for cross-modal recognition and the relevance of PLS to cross-modal face recognition. Theoretically, we have shown that in principle, there exist linear projections of images taken in two modalities that map them to a space in which images of the same individual are equal. This is true for images taken in different poses, at different resolutions, and approximately, for sketches and intensity images. Experimentally, we show that PLS and BLM can be used to achieve strong face recognition performance in these domains. Of particular note, we show that PLS has outperformed the best reported performance on the problem of face recognition with pose variation with impressive margin both in terms of accuracy as well as run-time and that Bilinear Models in all three domains outperformed many existing approaches. Moreover, using the exact same method we have also achieved comparable performance for sketch-photo and cross resolution face recognition.

Chapter 5

Pose-error Robust Discriminative Common Representation

In the previous chapter we have seen the use of Partial Least Square for learning a common representation of face images in different poses. This chapter investigates the performance of the PLS based common representation on datasets with more subjects and larger and less-controlled pose variation. Especially, we assess the performance of PLS based common representation in the presence of pose-errors. We show that pose-errors lead to degraded performance and describe a two-stage discriminative model to tackle them. The discriminative model is learned using simulated pose-errors and termed Adjacent Discriminative Multiple Coupled Latent Space or ADMCLS. We show the empirical benefits of ADMCLS over PLS based common representation for pose-invariant face recognition on two standard pose datasets: FERET and MultiPIE. This work was done in collaboration with Mourad Al Haj, Jonghyun Choi and Dr. Larry S. Davis; it resulted in the paper [108].

5.1 Motivation

A fully automatic real-world pose-invariant face recognition system involves various stages in the complete pipeline. Some important stages are - face detection, fiducial point estimation, pose estimation, alignment to a canonical pose and recognition or matching. The final recognition accuracy of the complete system depends on the accuracies of each of the stages. The entire pipeline leading up to recognition stage is still far from acceptable level of accuracy under real-life scenario. It calls for a systematic study of the effect of errors in various stages of the pipeline on the final performance and to develop built-in tolerance against small errors. Since the focus of this thesis is on learning common representation from multiple pose spaces, we carry out a detailed analysis of the effects of errors in the pose estimation on the performance of common representation based methods and also present some novel solutions to handle them. For the rest of this chapter, pose-error refers to the scenario when the *estimated/given* pose of a face is not the same as the *actual* pose. Note that even the ground-truth supplied with some of the databases may be wrong due to head movements of the subject during photo capturing.

5.2 Performance study with pose-error and more subjects

In this section, we first show the results of PLS based framework on FERET and MultiPIE datasets and discuss the reason behind the poor performance. Subsequently, we propose our extended two-stage discriminative approach followed by a detailed analysis of model parameters on the overall performance in later sections.

The performance of PLS based approach on two larger and less-controlled datasets (FERET and MultiPIE) is shown in Fig.5.6a and Fig.5.6b, respectively. From the figures it is evident that performance has decreased significantly for both MultiPIE and FERET. The most obvious reason is the increased number of testing subjects (gallery); FERET

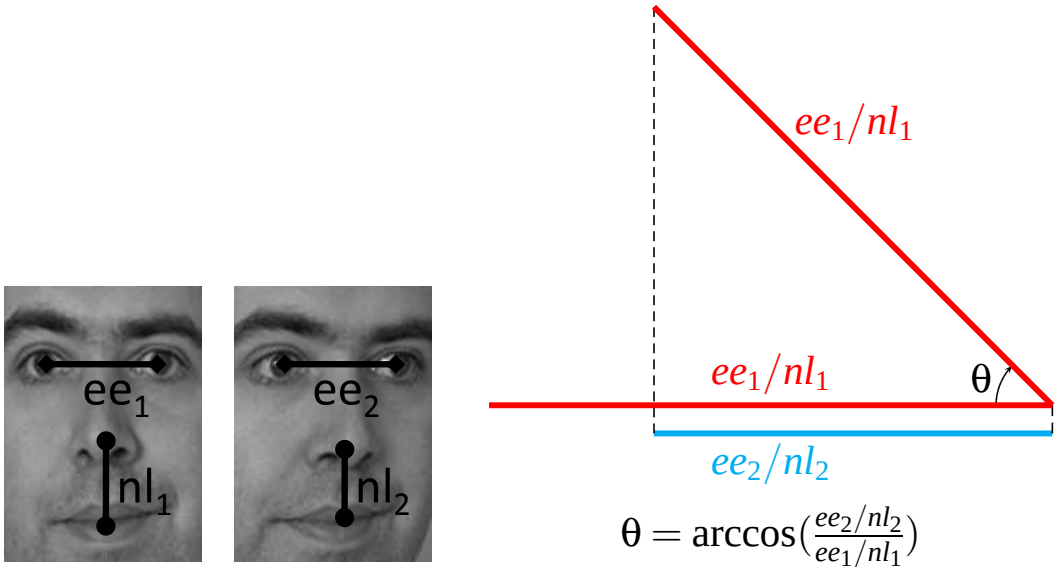
and MultiPIE have almost 3 and 7 times as many testing subjects as compared to CMU PIE, respectively. As the number of testing subjects increases, we need a discriminative representation for effective classification. All three i.e. CCA, BLM and PLS are generative in nature, hence, the decline in accuracy with increasing number of testing subject is natural. Secondly, we noticed that some of the faces in the dataset were off by a few degrees from the reported pose in the dataset. Especially for FERET, [11] has reported estimated poses which are very different from the ground-truth poses supplied with the dataset. Since projectors are learned using training images from FERET and MultiPIE, this leads to pose difference between the projectors and images. We term this phenomenon as *pose error*. It can occur because of head movement during acquisition or wrong pose estimation. Suppose, we learn two projectors for a $0^o/30^o$ gallery/probe pose pair. Let us assume that the 30^o testing images are not actually 30^o but $(30 \pm \theta)^o$ with $\theta \in [0, 15]$. For $\theta \leq 5$, the projectors and the testing images will have sufficient pixel correspondence. But for $\theta \geq 5$, we face the loss of correspondence, resulting in poor performance. Pose errors are inevitable and present in real-life as well as controlled conditions which is evident from FERET and MultiPIE. Moreover, due to different facial structures we may expect loss of correspondence for pose angles greater than 45^o. For example, both the eyes of Asians are visible even at a pose angle of around 60^o because of relatively flat facial structure as compared to European or Caucasian for which the second eye becomes partially or totally occluded at 60^o. This leads to missing facial regions at large pose angles which creates loss of correspondence. These pose errors become more frequent and prominent with increasing pose angles.

5.2.1 Pose estimation

In order to show that the poses provided in the FERET and MultiPIE databases are inaccurate, we assume that for each subject the frontal pose is correct and use this information to estimate the non-frontal poses; the change in the distance between the eyes of the subject, with respect to the distance in frontal pose, is used to calculate the new pose. In general, the change in the observed eye distance can be due to two factors: change in pose and/or change in the distance between the camera and the face. For the change in the face-camera position, the distance between the nose and the lip can be used to correct this motion, if present. For the pose change, in the two datasets, there is negligible change in yaw and the Euclidean distance automatically correct for any roll change, i.e. in-plane rotation; therefore, the Euclidean eye distance once corrected by the nose-lip distance can be directly used to measure the pitch pose.

The distance between the two eyes in frontal pose will be denoted by ee_1 and the distance between the nose and the lip by nl_1; similarly for the non-frontal pose to be estimated, the distance between the eyes is given by ee_2 and that between the nose and lip by nl_2. Assuming that the eyes, nose and lip are coplanar, i.e. the effect due to the nose sticking out is negligible, the new pose θ can be calculated as: $\theta = \arccos(\frac{ee_2/nl_2}{ee_1/nl_1})$. A pictorial demonstration of this calculation is shown in Fig.5.1.

To measure the poses in FERET and MultiPIE, manually annotated images were used to obtain the fiducial points and the frontal pose was used to calculate the rest of the non-frontal poses as explained above. The box and whisker plots for the estimated pose vs. the ground-truth pose for FERET and MultiPIE are shown in Fig.5.2a and Fig.5.2b, respectively. It is clear that, in both databases, there are inconsistencies between the different subjects at the same pose, rendering both ground truth data inaccurate.



(a) Fiducials (b) Schematic pose estimation

Figure 5.1: Schematic diagram to estimate the pose of a non-frontal face using fiducials.

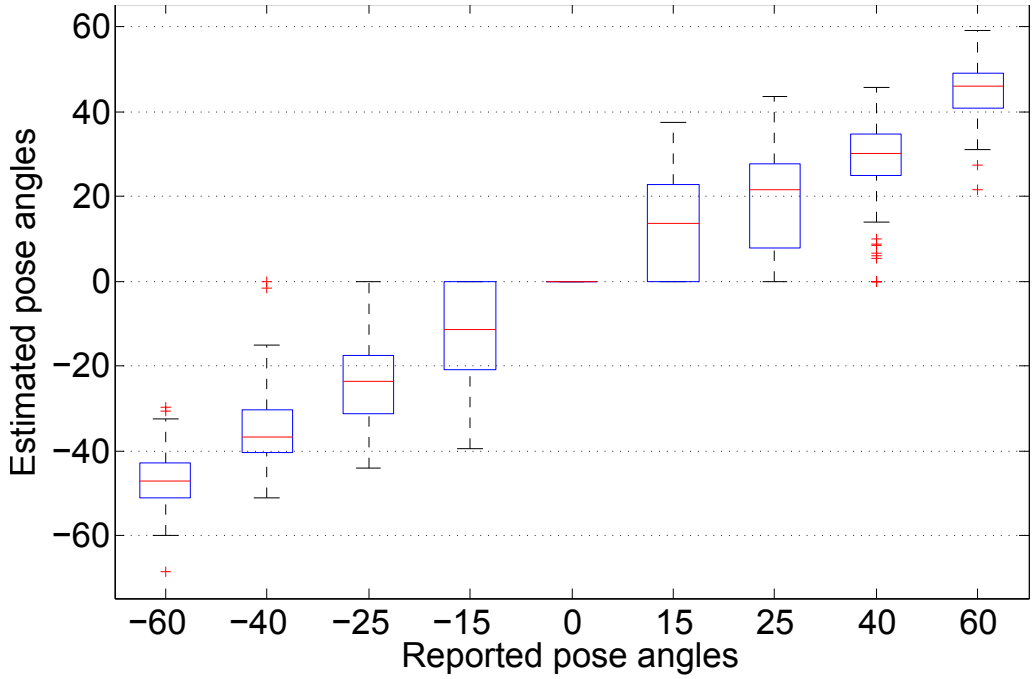
The pose errors are higher in magnitude and scatter in FERET which is obtained under unconstrained conditions as compared to MultiPIE.

5.2.2 Pose Estimation Tolerance

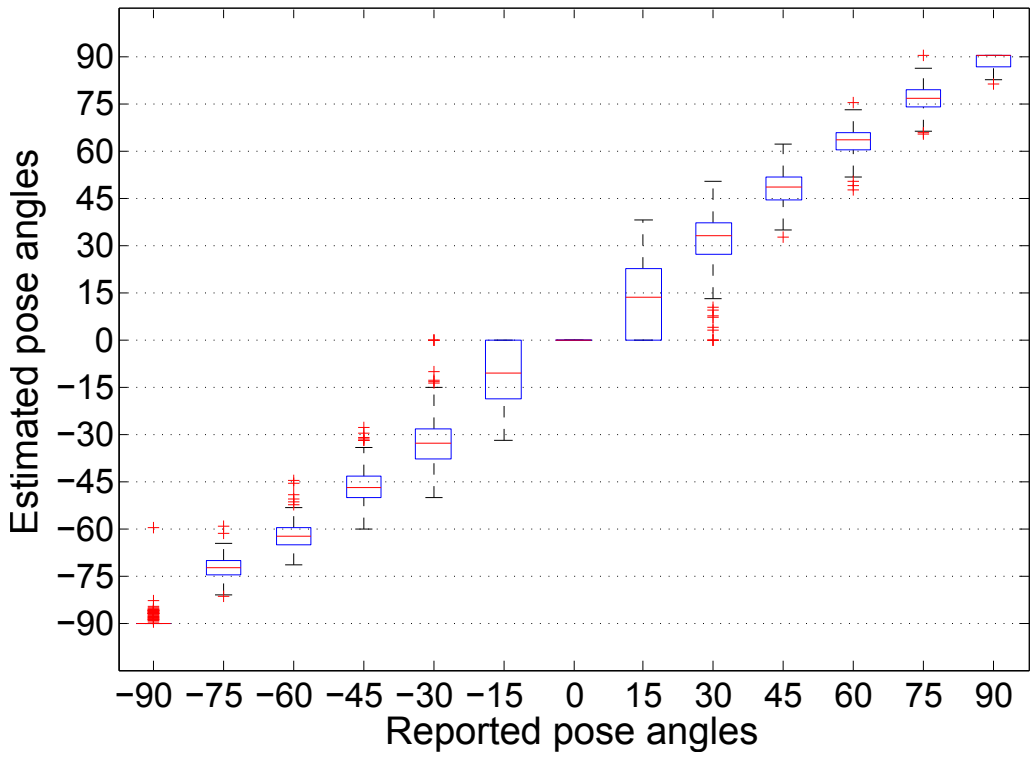
Human head pose could be estimated in various ways besides using fiducial locations. However, it is necessary to get a sense of robustness and accuracy of the approach for a reliable estimate. Therefore, we empirically estimate the sensitivity of fiducial-based-pose-estimation scheme. The accuracy of the estimated pose depends on the accuracy with which the fiducial points are located. Therefore, it is necessary to estimate the induced error in the estimated pose due to the errors in the fiducial points location. It is done by randomly perturbing all four fiducial locations and re-estimating the pose using the perturbed fiducial locations. The error is defined as the the absolute difference between the perturbed and originally estimated pose. The amount of perturbation for the eyes is a randomly chosen value between $\pm(x \times ee)$ i.e. fraction of the distance between the two eyes (ee). Similarly, nose and lips are perturbed by a randomly chosen value between $\pm(x \times nl)$ i.e. the same fraction of distance between the nose and lips (nl). The variation of average error over all the subjects and poses with increasing amount of perturbation fraction is shown in Fig.5.3. We can see that the error in pose estimation is increasing with the increment in the fiducial location error but it is not very high and only after an error of 15% in fiducial locations is the pose estimation severely affected.

5.3 Two-stage Discriminative Correspondence Latent Subspace

A discriminative representation approach such as LDA, requires multiple images per sample to learn the discriminative directions. We have a training set containing multiple images of a person but all the images are in different poses. Due to the loss of feature



(a) FERET pose error



(b) MultiPIE pose error

Figure 5.2: Box and Whisker plot for pose errors on FERET and MultiPIE data for all the poses which have only pitch variation from frontal.

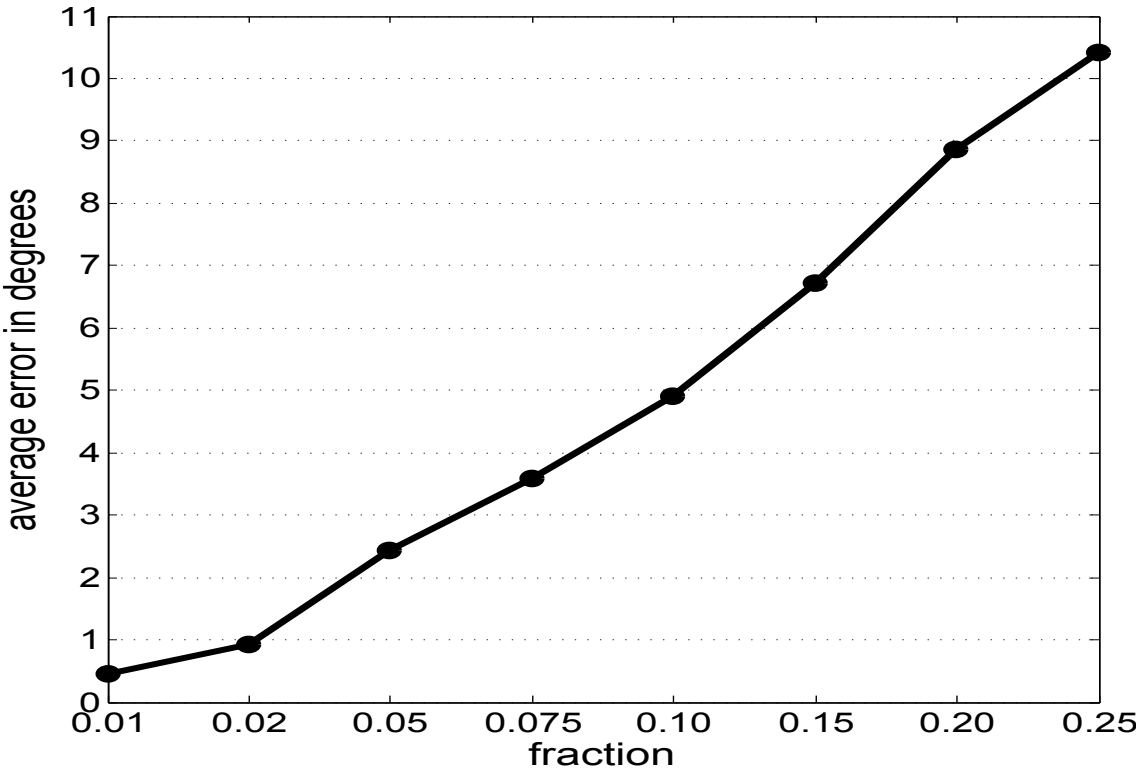


Figure 5.3: Variation of pose estimation error with the amount of random perturbation in the fiducial locations.

correspondence, we cannot use these multi-pose images directly to learn LDA directions. Results in [18] show that directly using them will lead to poor performance. However, we can learn a CLS for more than two poses simultaneously such that the projections of different pose images in the latent space have correspondence. Now, the multiple latent projections of a person can be used with LDA. Fortunately, using CCA as in (2.5), we can learn projectors for multiple poses to get a common CLS for a set of multiple poses. We empirically found that just using judiciously chosen set of poses (without LDA in latent space) to learn projectors offers some improvement over using only two poses. We defer the detailed discussion of selection of pose-sets and use of LDA to later sections. The multiple pose approach without LDA in latent space is termed Multiple CLS or MCLS and with LDA is termed Discriminative MCLS or DMCLS. The latent space projection $\mathbf{x}_l^i$ of i^{th} subject in pose p ($\mathbf{x}_p^i$) is given as

$$\mathbf{x}_l^i = W_p^T \mathbf{x}_p^i \tag{5.1}$$

Here, W_p^T is the projector for pose p and the subscript l indicates that $\mathbf{x}_l^i$ is in latent space. The projections of images in pose p using a projector for pose p are termed *same pose projections*. The latent space LDA offers discrimination based on the identity which is shown to be effective for classification [8, 125].

The performance drop study also suggests that pose error is an important factor and needs to be handled for better performance. To tackle the pose error, we draw motivation from [107, 126, 106] where it has been shown that the inclusion of expected variations (those present in the testing set) in the training set improves the performance.

Table 5.1: Framework names based on the components used, the super-script in the name denotes the CLS dimension. Abbreviations are - gal. is gallery; adj. is adjacent and Int. is Intermediate.

Name	Model	Training Set Poses	Projections	Classifier	CLS Dimension
CCA^{10}	CCA	gal. + probe	same pose	1-NN	10
PLS^{10}	PLS	gal. + probe	same pose	1-NN	10
BLM^{20}	BLM	gal. + probe	same pose	1-NN	20
$MCLS^{10}$	CCA	gal. + probe + Int.	same pose	1-NN	10
$DMCLS^{40}$	CCA	all poses	same + adj. pose	LDA	40
$ADMCLS^{10}$	CCA	gallery + probe + adj.	same + adj. pose	LDA	10

Specifically, [107] has shown that using frontal and 30^o training images with LDA improves the performance for 15^o testing images. And, [106] shows that using artificially misaligned images, created by small random perturbation of fiducial points in frontal pose, during training with LDA offers robustness to small errors in fiducial estimation. We combine the two approaches and artificially simulate pose errors. Unfortunately, creating small pose errors is not as simple as creating fiducial misalignment in frontal images. We do it by deliberately projecting face images onto adjacent pose projectors to obtain *adjacent pose projections*. The dataset used has pose angle increments in steps of 15^o; therefore, projection of a 45^o image onto 30^o and 60^o projectors will give adjacent pose projections for 45^o. The set of adjacent projections is given by

$$\mathcal{X}_l^i = \{\tilde{\mathbf{x}}_l^i : \tilde{\mathbf{x}}_l^i = W_{q\in A(p)}^T \mathbf{x}_p^i\} \tag{5.2}$$

here, $A(p)$ is the set of adjacent poses for pose p. The use of adjacent pose projections with LDA is expected to offer some robustness to small pose errors.

Same and adjacent pose projections have complementary information and both are important for robust pose-invariant face recognition. Therefore, we use both of them together as training samples with LDA to learn a discriminative classifier in the latent space. We call the resulting framework: Adjacent DMCLS of ADMCLS. ADMCLS is expected to offer robustness to pose errors smaller than 15^o which is indeed the general range of pose errors observed in real-life as well as controlled scenarios. Apart from providing robustness to pose error, adjacent projection also provides more samples per class for better estimation of class mean and covariance. We empirically found that inclusion of pose error projections dramatically improves the performance on FERET and MultiPIE which is in accordance with [106] and our intuition. It also supports our claim that performance drop is due to pose errors. The complete flow diagram for the ADMCLS framework is depicted in Fig.5.4.

5.3.1 Hyperparameter exploration

The proposed ADMCLS framework consists of two stages. The first stage involves learning the CLS and the second stage is learning the LDA directions using the projections in the latent subspace. Both stages have several different parameters, which will lead to different overall frameworks. For the ease of understanding and readability we summarize the names of different frameworks in Table 5.1. In this subsection we discuss the parameters involved and their effect on overall performance. We also discuss various criteria to choose these parameters and their effect on the final performance.

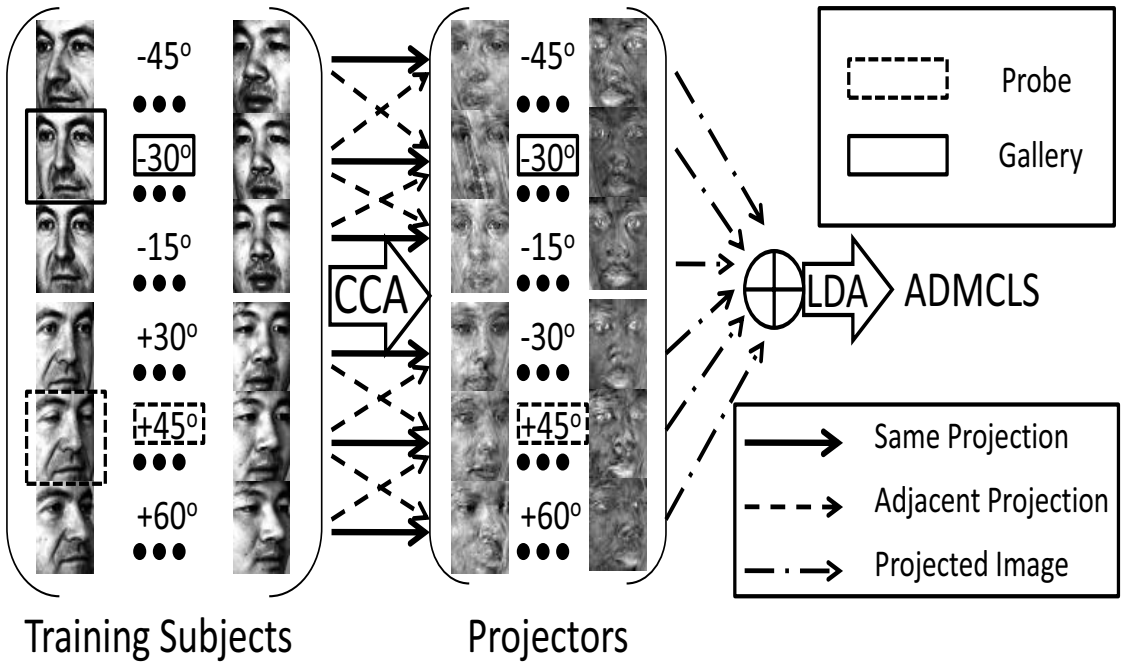


Figure 5.4: The flow diagram showing the complete ADMCLS process pictorially for a pair of gallery (-30^o) and probe ($+45^o$) pose pair. The gallery and probe along with adjacent poses constitute the set of poses for learning the CLS ($\pm 30^o$, $\pm 45^o$, -15^o and $+60^o$ for this case). Once the CLS is learned, same and adjacent pose projections (indicated by different arrow type) are carried out to obtain projected images in the latent subspace. An arrow from pose p images to pose q projector means projection of pose p images on pose q projector. All the projected images of a particular subject are used as samples in latent space LDA.

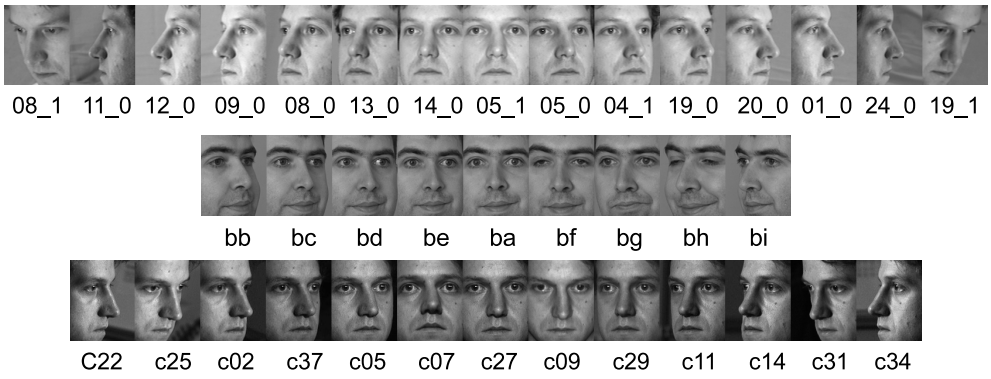


Figure 5.5: Images with pose names, **MultiPIE** (top row), **FERET** (middle row) and **CMU PIE** (bottom row).

To study the effect of a parameter, all the others were kept fixed but the one under study. Then the best values of individual parameters are used in the final framework. The final accuracy of the system in terms of rank-1 face identification rate is used as the performance measure to obtain the best value of each parameter. In order to facilitate future comparison of our approach, we have fixed the training subjects to be subject ID

1 to 34 for CMU PIE, 1 to 100 for MultiPIE and 1 to 100 (when arranged according to name) for FERET and made available the manually annotated fiducial points for FERET and MultiPIE used in our experiments. Testing is done on the rest of the subjects i.e. 34, 237 and 100 testing subjects for CMU PIE, MultiPIE and FERET respectively.

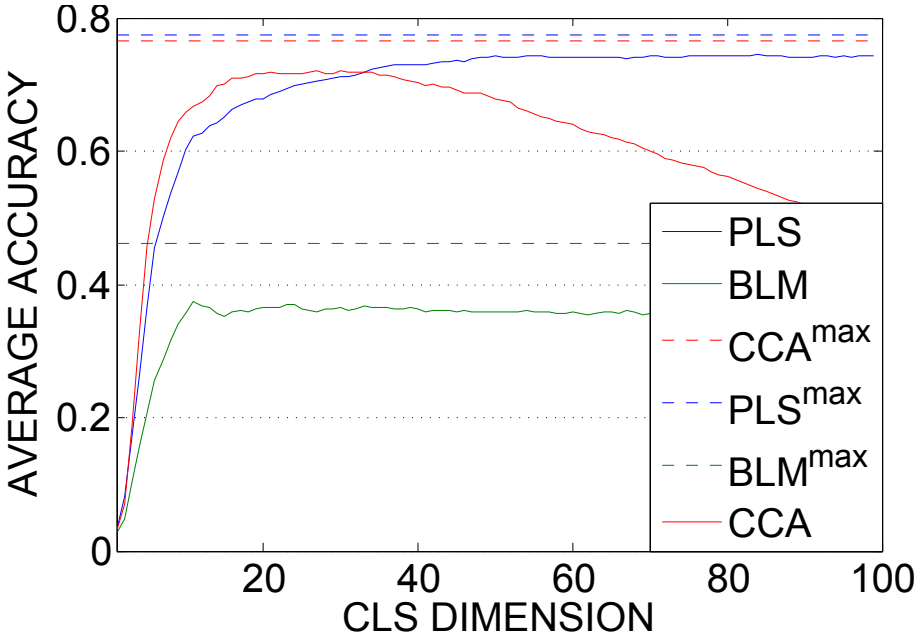
5.3.1.1 Latent Subspace Dimension and Learning Model

The subspace dimension is an important parameter in all the subspace based methods and plays a critical role in performance. Too many dimensions can lead to over-fitting and too few to under-fitting; therefore, this parameter needs to be decided very carefully. There are some techniques based on the spectral energy of the eigen-system that can guide the proper selection such as – choosing a pre-defined ratio of energy to be preserved in the selected number of dimensions – rejecting the directions with lower eigen-value than a threshold. In the case of CCA, we selected the top k eigen-vectors. We will see later that our final framework does not require a very careful selection of this parameter and is pretty robust to its variation. In the case of PLS we are using an iterative greedy algorithm and the number of dimensions can be selected by using only those directions which contain some pre-specified amount of total variation. However, it was observed that beyond a certain number of dimensions the accuracy remains constant. For BLM, we can use the spectral energy approach to select the number of dimensions. The selected number of dimensions of the CLS would be indicated as a superscript of the final framework name.

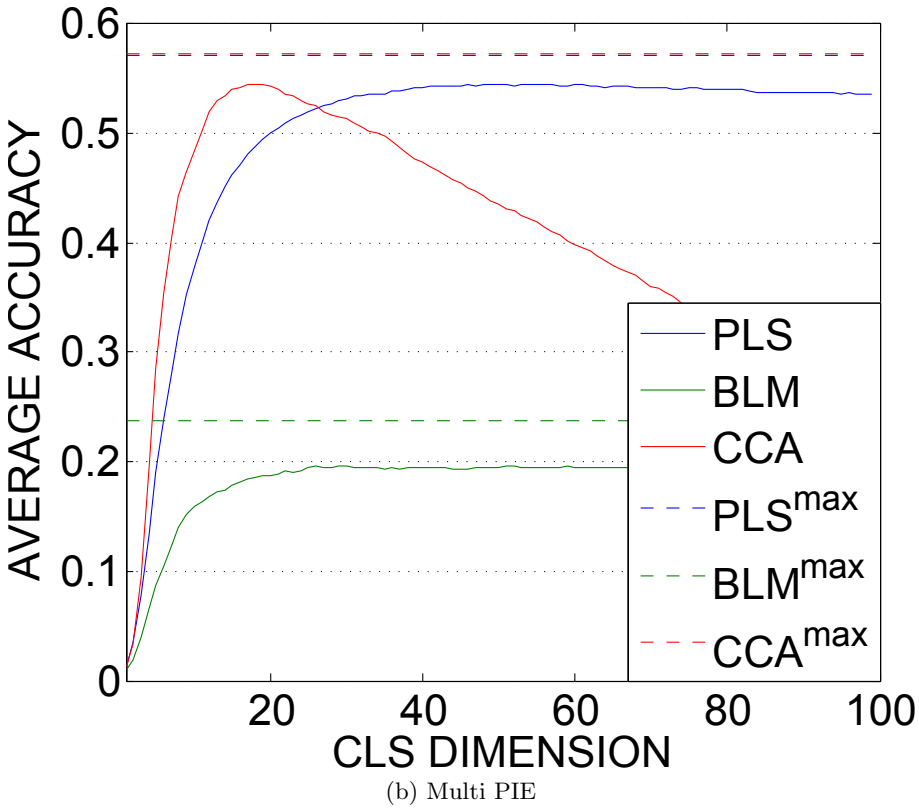
To keep things simple we have used 2 poses and 1-NN matching as the constituents of the final framework and varied the number of dimensions of CLS. The accuracy is the average accuracy for all possible gallery-probe pairs for the same number of CLS dimensions. There are 15 poses in MultiPIE so there is a total of 210 gallery-probe pose pairs and 72 for FERET (9 poses). The variation of accuracy for PLS, CCA and BLM on FERET and MultiPIE is shown in the Fig.5.6a and Fig.5.6b. It is obvious that different gallery-probe pairs will achieve the maximum accuracy with different number of CLS dimensions but we are calculating the average accuracy by considering the same CLS dimension for all pairs. To show the difference between our performance measure and the best possible accuracy obtained by using different CLS dimensions for different gallery-probe pairs, we calculated the best accuracy for all the pose pairs and averaged them to get the overall accuracy. These best accuracies are plotted as dashed horizontal lines in the same figure.

The choice of learning model has significant impact on the overall performance. We investigated three different choices for learning method: CCA, PLS and BLM and found that PLS performed slightly better than CCA for pose invariant face recognition and BLM is the worst performing [109]. However, PLS cannot be used to learn a CLS framework for more than two poses which makes it useless for the MCLS framework and BLM performs significantly worse than CCA. So, we used CCA for the cases when more than two poses are used for training.

Fig.5.6 clearly reveals the effect of learning model on face identification rate. The most important and satisfying observation is that the maximum possible accuracy is not significantly higher than the average accuracy justifying our assumption of equal CLS dimension across all gallery/probe pose pairs. Clearly, BLM performance is significantly worse than CCA and PLS which is in accordance with the results obtained in [109]. The performance of CCA and PLS is almost similar for MutliPIE and PLS performs better than CCA for FERET which is also in accordance with [109]. One clear observation from



(a) FERET



(b) Multi PIE

Figure 5.6: Result of CLS based recognition using 1-NN classifier on FERET and MultiPIE. (CCA/PLS/BLM)^{max} represents the maximum possible accuracy using different number of CLS dimensions for all gallery-probe pairs. For MultiPIE, PLS^{max} and CCA^{max} overlap and only one of them is visible.

the figure is that CCA performance is sensitive to CLS dimension and achieves maxima in a short range. On the other hand, the performance of BLM and PLS increase till a certain number of dimensions and then stays nearly constant. This brings out the fact that CCA is prone to overfitting while BLM and PLS are not.

5.3.1.2 Set of training poses

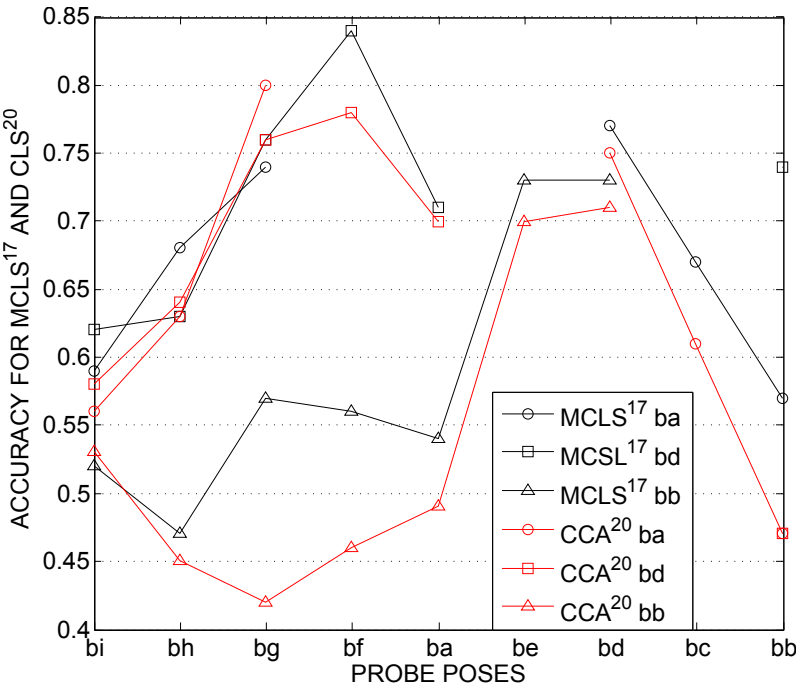
This has some effect on the obtained projectors since different sets of training poses will generate somewhat different projectors for each pose pair. Moreover, the supervised classifier in the latent space uses the projections as samples hence, it will have some bearing on the classifier too. In the case of PLS as the learning model, we can have only 2 training poses because of poor learning for multiple poses but this is not a problem with BLM or CCA. The set of poses used for training has deep impact on the obtained CLS performance and further improvements. We indicate the use of multiple training poses in the framework by preceding CLS by M i.e. MCLS.

The intuition of using more than two training poses can be understood in terms of robustness to noise offered by additional poses for CCA. It was pointed out and proved in [12] in a completely different context of clustering that adding more styles of data improves noise-robustness which also holds in our case of pose variation. As explained earlier in sub-section 2.3.2, CCA based CLS is a way of learning correspondence by maximizing correlation. The correlation between the training images in two different poses are most likely due to two factors: true correspondence and noise. We ideally want that the correlation is only due to correspondence. However, our data always contains some noise in the form of pose errors and/or inaccurate fiducial location. Presence of noise in the data can cause spurious correlations leading to false correspondence that will affect the performance. When more than two poses are used simultaneously, the obtained correlation between these poses has a higher probability of being due to correspondence because it is present in all the poses. However, this does not mean that we should add too many poses because it will decrease the flexibility of the learning model and lead to under-fitting. Thus, two poses will lead to over-fitting and too many will cause under-fitting, hence we choose four poses to strike a balance. Note that, the value four came out of empirical observation.

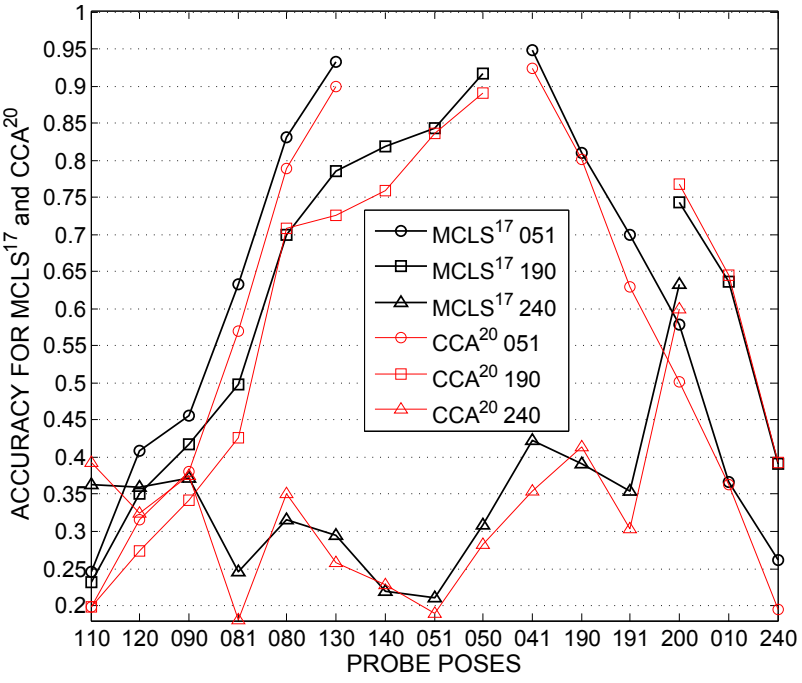
To evaluate the effect of changing the sets of training poses on the final framework for a particular gallery-probe pair, we include poses other than gallery and probe poses to learn CLS. This procedure raises some interesting questions: which poses should be included in training set? how many poses should be used? To answer these questions, we adopt a very simple approach that illustrates the effect of using multiple training poses. We use three gallery poses and all the possible probe poses for the selected gallery poses. For FERET, we choose pose **ba**(frontal), **bd** (25^o) and **bb** (60^o) and for MultiPIE, we choose 051(frontal), 190(45^o) and 240(90^o) as gallery poses. In addition to the gallery and probe we also select adjacent intermediate poses based on the viewing angle i.e. if we have gallery as frontal (0^o) and probe as $+60^o$ then we take two additional poses to be $+15^o$ and $+45^o$. Similarly, for gallery as frontal and probe as $+30^o$ we take only one additional pose $+15^o$ since it is the only intermediate pose.

Once the latent subspace is learned we use 1-NN for classification. The number of CLS dimensions is kept at 17 so the final frameworks are termed MCLS^{17}. We show the comparison of CCA based MCLS^{17} vs. CCA^{20} in Fig.5.8a and Fig.5.8b for FERET and MultiPIE respectively. There are some missing points in the performance curves in both

Figure 5.7: Projector bases corresponding to top eigen-values obtained using CCA (first 5 rows) and PCA (bottom 5 rows) obtained using 100 subjects from FERET. CCA projectors are learned using all the poses simultaneously and PCA projectors are learned separately for each pose. Each row shows the projector bases of the pose for equally indexed eigen-value. Observe that, projector bases are hallucinated face images in different poses and the CCA projector bases look like rotated versions of the same hallucinated face but there is considerable difference between PCA projectors. This picture visually explains the presence of correlation in the latent CLS space using CCA.



(a) FERET



(b) Multi PIE

Figure 5.8: Comparison of $MCLS^{17}$ vs. CCA^{20} with varying gallery-probe pairs for a) three gallery poses ba(frontal), bd(40^o) and bb(60^o) on FERET dataset. b) Three gallery poses 051(frontal), 190(45^o) and 240(90^o) on MultiPIE dataset. $MCLS^{17}ba$ indicates that the gallery is pose **ba**, **multiple** poses are used during training and CCA is the learning model with **17** dimensional CLS and **1-NN** classifier while $CCA^{20}ba$ indicates that the gallery is pose **ba**, **two** poses are used during training and **CCA** is the learning model with **18** dimensional CLS and **1-NN** classifier

figures because an adjacent gallery-probe pose pair does not have any intermediate pose. The comparison clearly highlights the improvement offered by using multiple poses for learning the latent subspace. We generally observe some improvement with the MCLS[17] framework for gallery and probe poses with large pose difference except for few places where it either remained the same or decreased slightly. We also observe that the improvement is more significant in FERET as compared to MultiPIE which is due to the fact that MulitPIE dataset has less pose errors than FERET, as shown in subsection 5.2.1. Therefore, MCLS framework has more to offer in terms of robustness to pose errors in FERET as compared to MultiPIE.

The second stage of the framework is learning a supervised classifier using the latent subspace projections. This stage has two crucial parameters: Set of projections and classifier. The next two sections explore their affect on the performance.

5.3.1.3 Set of projections and Classifier

This subsection explores the combination of the set of latent subspace projections for a subject and the classifier used for matching. As discussed earlier, we have two choices for projecting a face image in the CLS and both contain complementary information which can be utilized by a classifier for recognition. Since all the databases used in this paper have pose angles quantized in steps of 15^o, the difference between any two adjacent poses is 15°. In our framework, we do not consider more than 15^o pose difference because they will render the projection meaningless and they do not exist in real life scenarios.

As mentioned earlier, CCA is used as the learning model for all the experiments with more than two poses in the training. MultiPIE has 15 poses and FERET has 9, so the size of the eigen-system for MultiPIE becomes too big and requires large memory. So, all the exploratory experiments were done with FERET and conclusions were used to decide the optimal strategy for MultiPIE. In order to avoid under-fitting we adopt a simple strategy to select a subset of poses for training that is based on gallery-probe pair. The gallery-probe pairs along with the adjacent poses of them are selected as the training set of poses. So, for a $+45^o/-30^o$ gallery/probe pair the training set would be $\pm 30^o, \pm 45^o, +60^o, -15^o$ and for $-15^o/0^o$ training pose set is $\pm 15^o, 0^o, 30^o$. Adjacent poses are selected to simulate pose error scenario. We call this variant of DMCLS as Adjacent Discriminant Multiple Coupled Subspace (ADMCLS). To evaluate the effect of different latent space projections, we plot the average accuracy across all 72 gallery/probe pairs in Fig.5.9 for the following settings: 1-NN classifier with two poses denoted by CLS; Intermediate poses and 1-NN classifier denoted by MCLS; two poses and LDA denoted by DCLS; all 9 poses for FERET and adjacent projections with LDA denoted by DMCLS and adjacent set of training poses with adjacent projections and LDA denoted by ADMCLS.

It is clear from the Fig.5.9 that ADMCLS performs the best closely followed by DMCLS, while, CLS is the worst performing approach with DCLS and MCLS performance being slightly better than CLS. The use of LDA with adjacent projections did not only increase the accuracy significantly but also makes the final framework fairly insensitive to CLS dimension, which eliminates the burden of determining it by cross-validation. This significant improvement is due to artificial simulation of pose error scenarios and learning to effectively neglect such misalignments for classification using LDA. One more reason contributing to the improvement is the LDA assumption of similar within-class covariance for all the classes. In our case, indeed the within-class covariance matrices are almost the same because the samples of all the classes in CLS are obtained using the same set of CLS

bases and the types of projection are also the same for all the classes. The recognition rates for all the 72 pose pairs with DMCLS[40] using all the pose pairs in training set are given in Table 5.2.

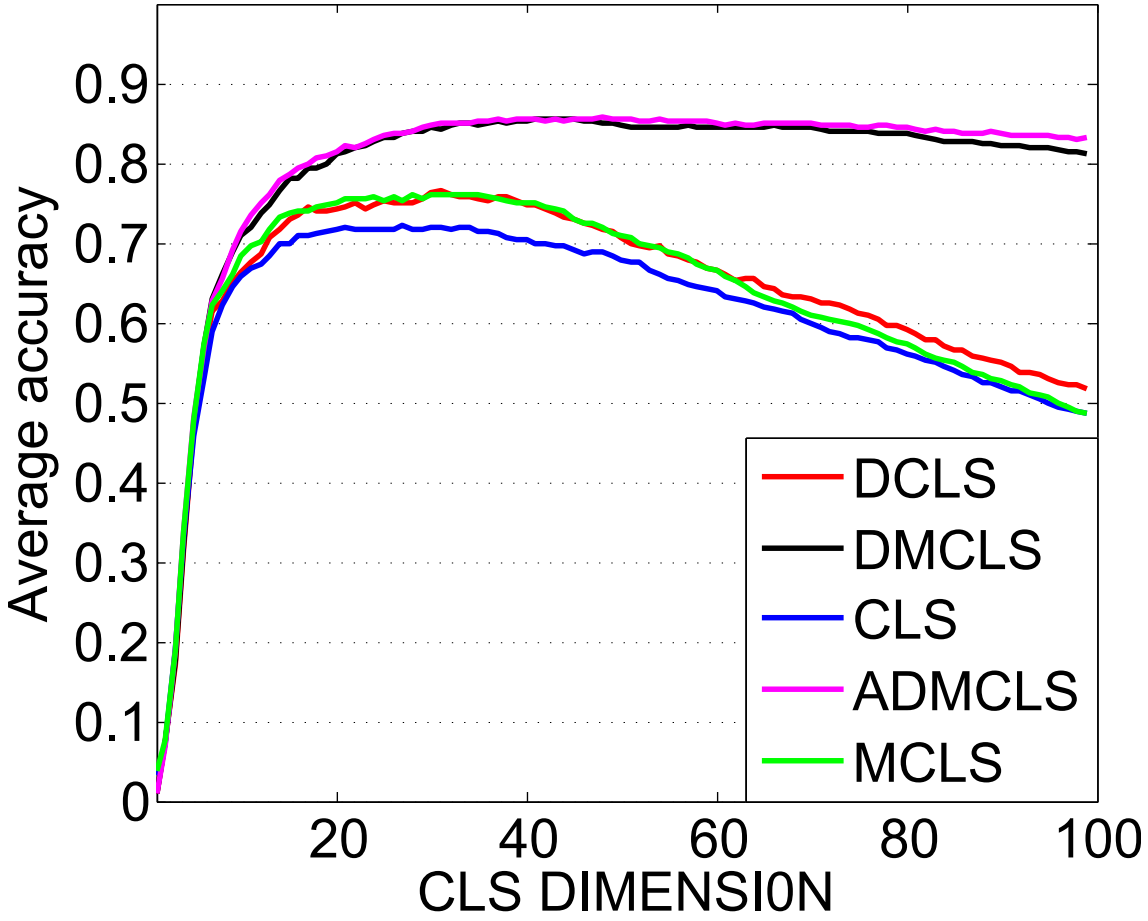


Figure 5.9: Variation of CLS, MCLS, DCLS, DMCLS and ADMCLS accuracy with latent space dimension for all the gallery-probe pairs on FERET.

To prove the point that the improvement is actually due to handling pose errors we also obtain the relative improvement by ADMCLS[40] over CLS[22] for all gallery-probe pairs. The difference is plotted as a heat map for better visualization in the Fig.5.10a. From the figure, it is evident that the most significant improvements are in the cases where either the gallery or the probe pose is far away from frontal pose. In these cases, the occurrence and severity of pose errors and incorrect fiducial locations is most likely and prominent.

5.3.2 Computational Complexity

It is obvious that learning an ADMCLS with multiple poses offers various advantages but it also requires some additional computational cost. The computational bottleneck of the ADMCLS framework is the solution of the generalized eigen-value problem in (2.5). The complete generalized eigen-value decomposition of a pair of $N \times N$ square matrices (A, B) is $O(N^3)$ but we only need the leading k eigen-vectors. Therefore, the cost comes down to $O(kN^2)$. In our case, $N = \sum_m D_m$ where, D_m is the dimension of the m^{th} pose feature space (number of pixels in our experiments). For simplicity, let's assume that the

Table 5.2: $DMCLS^{40}/ADMCLS^{40}$ for all possible gallery-probe pairs on FERET

Pose Angle	bi -60^o	bh -40^o	bg -25^o	bf -10^o	ba 0^o	be 10^o	bd 25^o	bc 40^o	bb 60^o	DMCLS^{40} Avg/ ADMCLS^{40} Avg
bi	-/-	98/98	92/93	88/82	70/77	81/80	79/80	76/69	70/63	**81.75**/80.25
bh	97/97	-/-	99/99	94/94	80/84	90/87	79/77	71/70	62/60	**84.00**/83.50
bg	95/96	97/99	-/-	100/100	91/92	98/97	90/92	78/76	68/68	89.63/**90.00**
bf	83/91	93/95	96/99	-/-	93/97	97/99	95/95	85/84	73/71	89.38/**91.37**
ba	75/79	77/85	89/94	91/96	-/-	90/95	87/94	81/82	67/70	82.13/**86.38**
be	86/83	91/88	96/96	98/99	90/99	-/-	99/100	97	84	92.50/**93.25**
bd	79/78	84/83	90/90	91/95	90/89	98/98	-/-	98	84/86	89.25/**89.63**
bc	75/70	73/67	77/73	82/79	80/80	92/94	97/97	-/-	95/96	**83.88**/82.00
bb	71/70	66/60	67/62	67/67	64/65	81/82	82/84	95/95	-/-	**74.13**/73.12

dimension of each pose feature space is equal to a constant D. Therefore, $N = MD$, where M is the number of coupled poses. Hence, the computational complexity as a function of the number of coupled poses M and the dimension of feature space is $O(kD^2M^2)$.

5.4 Experimental Analysis

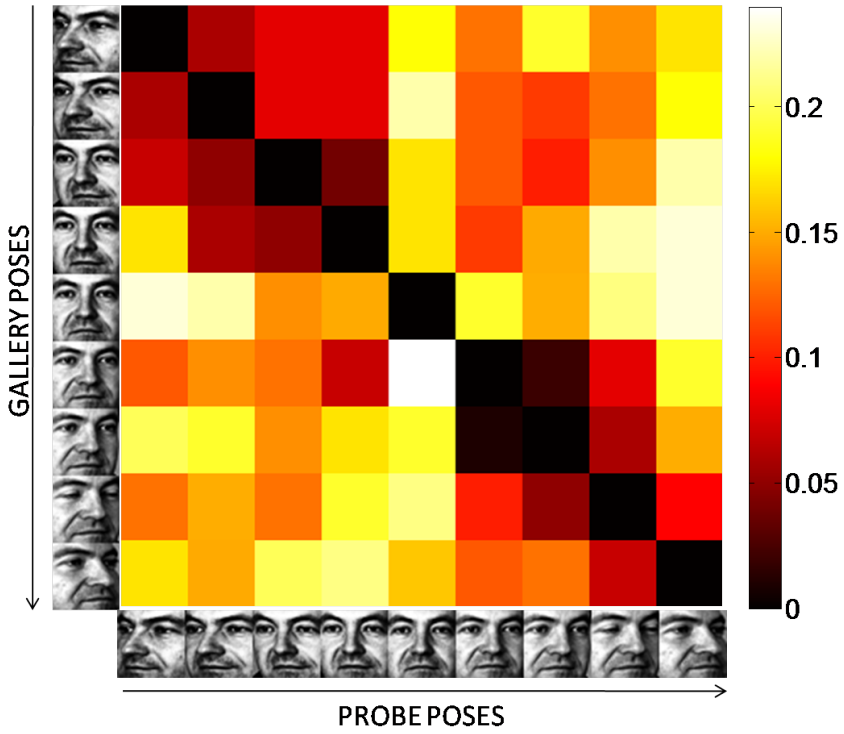
In this section we provide the rank-1 indentification rates obtained on CMU PIE, FERET and MultiPIE using best parameters settings and compare our results with prior work on the same datasets. Please note that, CCA is used as the learning model for all the methods using more than two poses in training set, for the reasons explained in previous sections.

5.4.1 Training and Testing Protocol

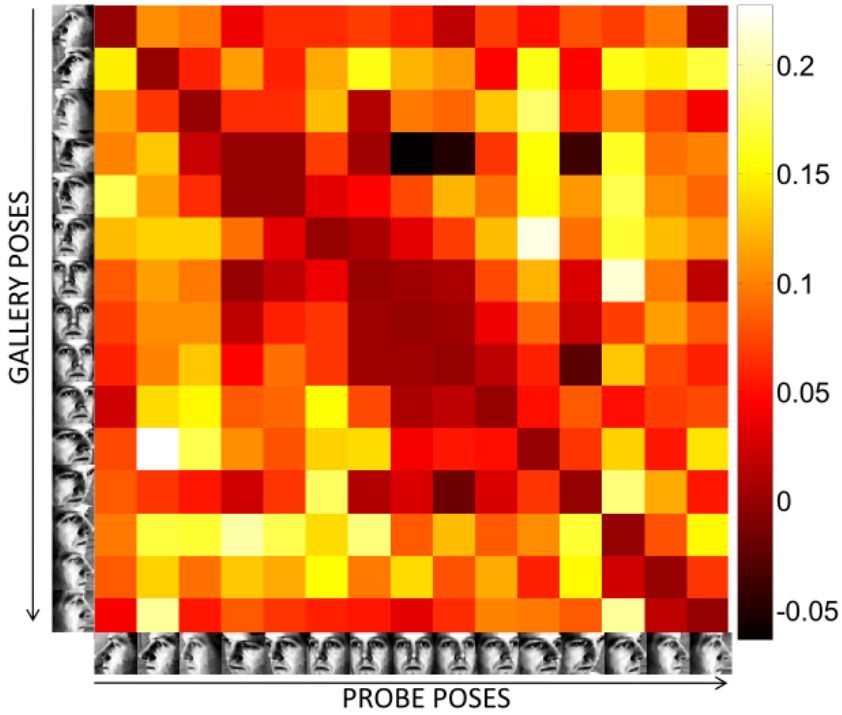
Like any other learning based approach we require training data to learn the model parameters. We assume access to a training data that has multiple images of a person under different poses and ground-truth poses of training as well as testing faces. Although fiducial points can be used for a better estimation of pose, we use the ground-truth poses for a fair comparison with previous approaches. Moreover, automatic pose estimation algorithms and fiducial detectors always have some error. Therefore, working with small pose errors reflects performance with automatic pose or fiducial detector. CMU PIE, FERET and MultiPIE have multiple images of a person under a fixed set of poses. Hence, we use some part of the data as training and the rest as testing. We also need to align the faces under different poses which requires fiducial landmark points. In the training phase, we obtain the projectors for all the possible gallery/probe pose pairs for the required framework i.e. ADMCLS, DMCLS etc. At testing time, we assume that the gallery and probe poses are known and use appropriate projectors for projection followed by matching. For testing purpose we always project the images on the same pose projector as per as the ground-truth poses. For a completely automatic face recognition system, pose and fiducial landmarks should be obtained automatically. However, for experimentation purposes, we assume them to be known beforehand, a common practice followed in much previous work [91, 92, 65, 53, 35, 5, 78, 64, 107, 66]. Prima facie, it may look like a serious limitation but research and commercial systems have shown impressive performance in automatic pose and fiducial determination [9, 15] that can be used in conjunction with our approach to make an automatic pose invariant face recognition system.

Table 5.3: comparison of ADMCLS^{40} with other published works on feret with frontal gallery.

Method	Probe pose								
	bi	bh	bg	bf	be	bd	bc	bb	Avg
LDA [64]	18.0	55.0	78.0	95.0	90.0	78.0	48.0	24.0	60.8
LLR [64]	45.0	55.0	90.0	93.0	90.0	80.0	54.0	38.0	68.1
CCA [64]	65.0	81.0	93.0	94.0	93.0	89.0	80.0	65.0	82.5
Stack [5]	40.0	67.5	88.5	**96.5**	94.5	86.0	62.5	38.0	71.7
Yamada [53]	8.5	32.5	74.0	88.0	83.0	54.0	23.5	6.5	46.3
Ridge+Int [65]	67.0	77.0	90.0	91.0	92.0	89.0	78.0	69.0	81.6
DMCLS^{40}	75.0	77.0	89.0	91.0	90.0	87.0	81.0	67.0	82.1
ADMCLS^{40}	**79.0**	**85.0**	**94.0**	96.0	**95.0**	**90.0**	**82.0**	**70.0**	**86.4**
3DMM [11]	*90.7*	*95.4*	*96.4*	*97.4*	*99.5*	*96.9*	*95.4*	*94.8*	*95.8*
Ridge+Gab [65]	*87.0*	*96.0*	*99.0*	*98.0*	*96.0*	*96.0*	*91.0*	*78.0*	*92.6*
3DMM-LGBP [6]	–	*90.5*	*98.0*	*98.5*	*97.5*	*97.0*	*91.9*	–	*95.6*



(a) FERET improvement map



(b) MultiPIE improvement map

Figure 5.10: Improvement map for (a) using ADMCLS[40] over CCA[20] for FERET and (b) using ADMCLS[25] over CCA[18] for MultiPIE. The original accuracies were all between 0 (0%) and 1 (100%). It is evident from the two maps that the amount of improvement is more in FERET as compared to MultiPIE. Also, the improvement is more when either the gallery or probe pose is far from the frontal view.

Table 5.4: MultiPIE accuracy for all possible 210 gallery-probe pairs using ADMCLS[25] with 237 testing subjects. The duplet below the pose name indicates the horizontal, vertical angle i.e. 45,15 means 45^o horizontal and 15^o vertical angle.

Prb→ Gal↓	110 -90,0	120 -75,0	090 -60,0	081 -45,45,	080 -45,0	130 -30,0	140 -15,0	051 0,0	050 15,0	041 30,0	190 45,0	191 45,45	200 60,0	010 75,0	240 90,0	**Avg**
110	-/-	76.4	65.8	34.6	48.5	37.6	33.3	27.4	21.9	31.6	31.2	24.9	35.9	49.4	43.9	**37.5**
120	78.5	-/-	81.9	48.5	68.8	57.8	54.9	43.9	42.2	44.7	44.7	27.4	59.1	65.0	50.2	**51.2**
090	67.1	81.9	-/-	59.5	80.2	72.2	51.9	46.0	46.8	54.0	55.3	32.1	64.1	60.8	43.0	**54.3**
081	38.0	49.8	57.8	-/-	78.5	82.3	73.8	55.7	48.9	52.3	57.0	63.7	49.8	40.1	28.7	**51.8**
080	55.3	70.9	78.9	76.8	-/-	97.9	93.2	85.7	84.8	82.7	84.0	54.0	72.6	59.9	40.1	**69.1**
130	39.7	58.6	72.6	84.4	97.0	-/-	96.2	93.7	92.8	90.7	86.9	60.8	68.4	54.9	33.8	**68.7**
140	30.4	52.7	57.0	73.8	90.7	97.5	-/-	98.7	95.4	92.8	89.0	60.8	64.1	45.6	24.1	**64.8**
051	27.0	42.2	48.5	58.6	84.8	96.6	99.2	-/-	99.2	96.2	89.0	65.0	57.4	47.7	27.8	**62.6**
050	25.7	40.9	47.7	54.0	85.2	95.4	97.5	98.7	-/-	98.7	94.9	74.7	75.1	59.5	35.9	**65.6**
041	26.6	50.2	51.9	52.3	81.0	93.7	95.8	94.9	98.7	-/-	96.6	88.6	80.6	72.6	43.9	**68.5**
190	27.4	50.2	51.9	53.2	78.9	86.1	89.9	87.8	94.5	97.5	-/-	85.7	90.3	70.0	53.6	**67.8**
191	22.8	30.8	30.8	65.0	49.8	65.8	60.8	62.4	70.0	87.3	83.1	-/-	77.2	63.3	39.2	**53.9**
200	36.3	59.1	65.8	52.3	72.2	67.9	63.7	58.6	72.2	84.4	87.3	81.0	-/-	97.0	75.1	**64.9**
010	44.7	63.7	61.6	43.0	64.6	53.2	47.7	54.0	63.7	77.6	75.5	65.4	95.4	-/-	94.9	**60.3**
240	43.5	52.3	43.0	26.6	41.8	31.6	28.3	22.4	34.6	45.6	51.1	38.8	79.7	93.2	-/-	**42.2**

5.4.2 FERET

This dataset contains 200 subjects in 9 different poses spanning $\pm 60^o$ view-point. All the images for one person along with the pose name are shown in Fig.5.5. Pre-processing steps similar to CMU PIE were used except that the final facial region crops are of size 50×40 pixels. Subjects 1 to 100 were chosen as training subjects and 101 to 200 as testing. Since, there are 9 poses, we have 72 different gallery-probe pairs.

We report the accuracy for FERET data set using two different variants of DMCLS to bring out the fact that using more than the required number of poses in training may lead to poor performance. We report DMCLS based accuracy which uses all the 9 poses in the training and adjacent projection based LDA in latent space and ADMCLS based accuracy which uses a subset of poses for training. The number of CLS dimension is indicated as the superscript and CCA is used as the learning model. Table 5.2 reports the accuracy for all possible gallery-probe pairs using the two different variant i.e. DMCLS and ADMCLS. The table clearly indicates the advantage of using ADMCLS over DMCLS when near frontal poses are used as gallery pose. It also indicates that when extreme poses are gallery then using DMCLS is slightly better than ADMCLS, a possible explanation is that extreme poses require more regularization than flexibility. We report the accuracy obtained using 3DMM [11] approach to indicate the performance difference between 2D and 3D approaches. The difference in performance between 2D and 3D approaches supports the fact that 3D information improves performance in pose invariant face recognition.

The results of [65] are shown under two settings: with and without Gabor features. The authors have extracted Gabor features at 5 hand annotated fiducial locations using 5 scales and 8 orientations resulting in 200 local classifiers which they fuse using the technique given in [53]. The method involves modeling the conditional probability of the Gabor response g_i of classifier i for same and different identities i.e. $P(g_i|same)$ and $P(g_i|dif)$ respectively. Then, Bayes Rule is used to obtain posteriors $P(same|g_i)$ and $P(dif|g_i)$ and the probability of final classification is the sum of the posterior probabilities. The inclusion of Gabor features has improved the accuracy dramatically because they are more discriminative than intensity features. Moreover, using Gabor features at hand-annotated fiducial landmarks is providing manual correspondence to the learning method. Combining Gabor features with probabilistic fusion is interesting and worth trying within our framework. Surprisingly, for CMU PIE our simple PLS based approach even outperformed the Gabor feature based approach.

5.4.3 Multi PIE

MultiPIE is an extension of CMU PIE data set containing more subjects and more pose-variation. It has a total 337 subjects photographed in 4 different sessions, under 15 different poses, 20 illumination conditions and 4 different expressions. We only took neutral expression and frontal lighting images for our experiments. All the pre-processing steps are the same as in CMU PIE except that the cropped facial region is 40×40 pixels. We took subject ID 1 to 100 as training and 101 to 346 as testing, resulting in a total of 237 testing subjects. For MultiPIE we could not obtain MCLS using all the poses in the training set due to memory problem associated with large eigen-value problem. Hence, we adopt the ADMCLS approach to select a subset of training poses and report the accuracy in Table 5.4. The MultiPIE data is relatively new and not many results are reported for pose invariant face recognition on it. We show our results along with the results of other

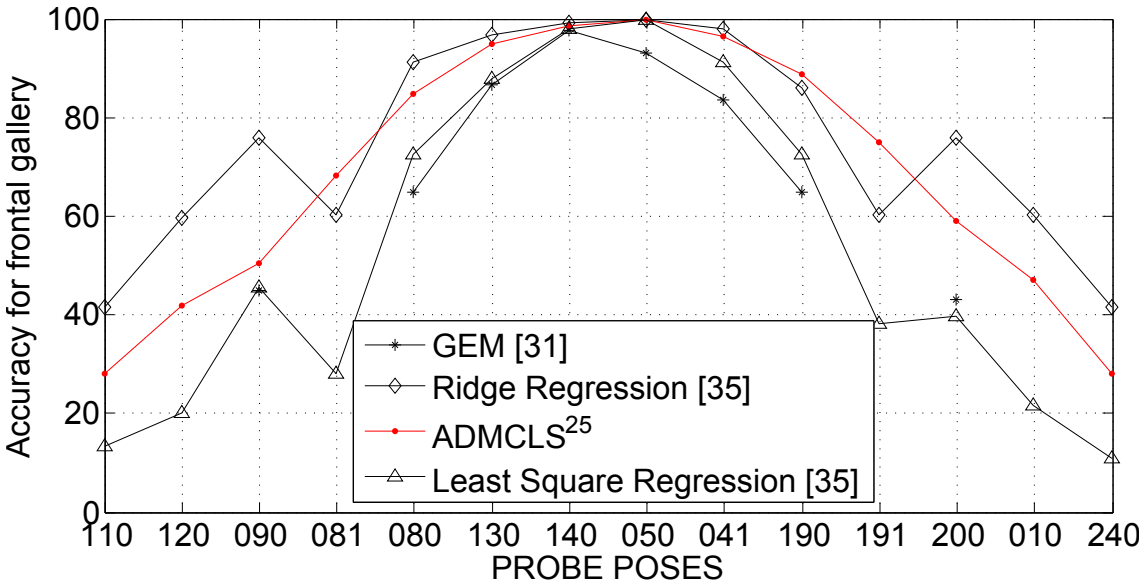


Figure 5.11: Comparison of $ADMCLS^{25}$ with other approaches on MultiPIE dataset with frontal gallery.

works in Fig.5.11. It should be noted that we are reporting the results of [65] with pixels intensities as feature.

Interestingly, our 2D approach is better than the 3D GEM [90] approach. We also observe that our approach is comparable to the approach in [65] for small pose differences but the difference increases with the pose angle. This might be due to the fact they report their result under frontal gallery and non-frontal probe only, giving them the opportunity to better tune the parameter but we report the results under general pose variation and do not optimize our method for frontal gallery and non-frontal pose. Moreover, we have outperformed [65] on both CMU PIE and FERET by large margins without optimizing for the case of frontal gallery images.

5.5 Conclusion and Discussion

We have proposed a generic Discriminative Coupled Latent Subspace based method for pose invariant face recognition. The learned set of coupled subspaces projects the images of the same person under different poses to close locations in the latent space, making recognition possible using a simple 1-NN or discriminative learning. We have discussed the conditions for such projection directions to exist and perform accurately. We further exploit the property of CCA to couple more than two subspaces corresponding to different poses and show that judiciously using multiple poses to learn the coupled subspace performs better than using just two poses. That is because information from multiple views is more consistent and robust to noise (pose errors and incorrect fiducials) than just two views. Multiple coupled subspaces also provide us with the opportunity to generate multiple samples of a person in the latent subspace which can be used with LDA to encode discriminative information.

We have provided empirical evidence that pose-invariant-face recognition suffers from pose errors even under controlled settings, leading to poor performance. We tackle

this problem by artificially simulating pose error scenarios via adjacent-pose-latent projection. The latent projections obtained by projecting the images of a person under different poses on the same and adjacent pose projectors are used with LDA to effectively avoid the drop in performance due to small pose errors. The proposed approach has achieved state-of-the-art results on CMU PIE and FERET when four fiducial points are used with simple intensity features and comparable results on MultiPIE.

We experiment with pose variation only and illumination is considered to be constant. However, owing to the independent block structure of the overall framework, it can be easily extended to handle lighting variations by using some illumination invariant representation such as: The Self Quotient Image [134], Oriented gradient [46] etc... Moreover, Gabor features extracted at specific fiducial locations can be used to improve the performance further as in [65, 91, 92, 66, 6]. The coupled subspaces are learned in generative manner and only after projection on these subspaces, label information is used with LDA. The method could be improved by learning a discriminative coupled subspace directly. Learning such a subspace and using it for pose and lighting invariant face recognition is one of our future endeavors.

Chapter 6

Generalized Multiview Analysis

The previous chapter discussed the use of a two-stage discriminative model for pose-invariant face recognition, however, it did not tackle the problem of pose and lighting variation simultaneously. This chapter presents a general multi-view (*view* has the same meaning as *modality* throughout this chapter) content extraction approach that we call Generalized Multiview Analysis or GMA. It affords simultaneous learning of common representation and discriminative projections. It has all the desirable properties required for cross-view classification and retrieval: it is supervised, it allows generalization to unseen classes, it is multi-view and kernelizable, it affords an efficient eigenvalue based solution and is applicable to any domain. GMA exploits the fact that most popular supervised and unsupervised feature extraction techniques are the solution of a special form of a quadratic constrained quadratic program (QCQP), which can be solved efficiently as a generalized eigenvalue problem. GMA solves a joint, relaxed QCQP over different feature spaces to obtain a single (non)linear subspace. Intuitively, GMA is a supervised extension of Canonical Correlational Analysis (CCA), which is useful for cross-view classification and retrieval. The proposed approach is general and has the potential to replace CCA whenever classification or retrieval is the purpose and label information is available. We outperform previous approaches for text-image retrieval on Pascal and Wiki text-image data. We report state-of-the-art results for pose and lighting invariant face recognition on the MultiPIE face dataset, significantly outperforming other approaches.

6.1 Motivation

The motivation comes from the fact that utilization of label information while learning the common subspace can improve the performance over unsupervised learning when the task is classification. This can be visualized by looking at the toy example in Fig 6.1. It's a simple pictorial demonstration of various multi-view approaches along with the proposed GMA and an ideal approach. Shapes represent classes, the same color and shape indicates paired samples in different views, dashed outline shapes (triangles) are the unseen classes (not used in training). Ideally, we would like different classes (seen and unseen) to be well separated with all the same-class samples collapse to a point. Unsupervised approaches like CCA, PLS and BLM try to unite paired samples only. Supervised approaches, like SVM-2K and HMFDA unite same-class samples and separate different classes but they cannot generalize to unseen classes. Our proposed approach GMA, unites same class samples, separates different classes and generalizes to unseen classes.

6.2 Related work

As discussed earlier in Sec. 2.3, the popular approaches to learn common latent subspace are Canonical Correlational Analysis (CCA) [39, 38], Bilinear Model (BLM) [128] and Partial Least Squares (PLS) [99, 39, 109]. Specifically, CCA has been the

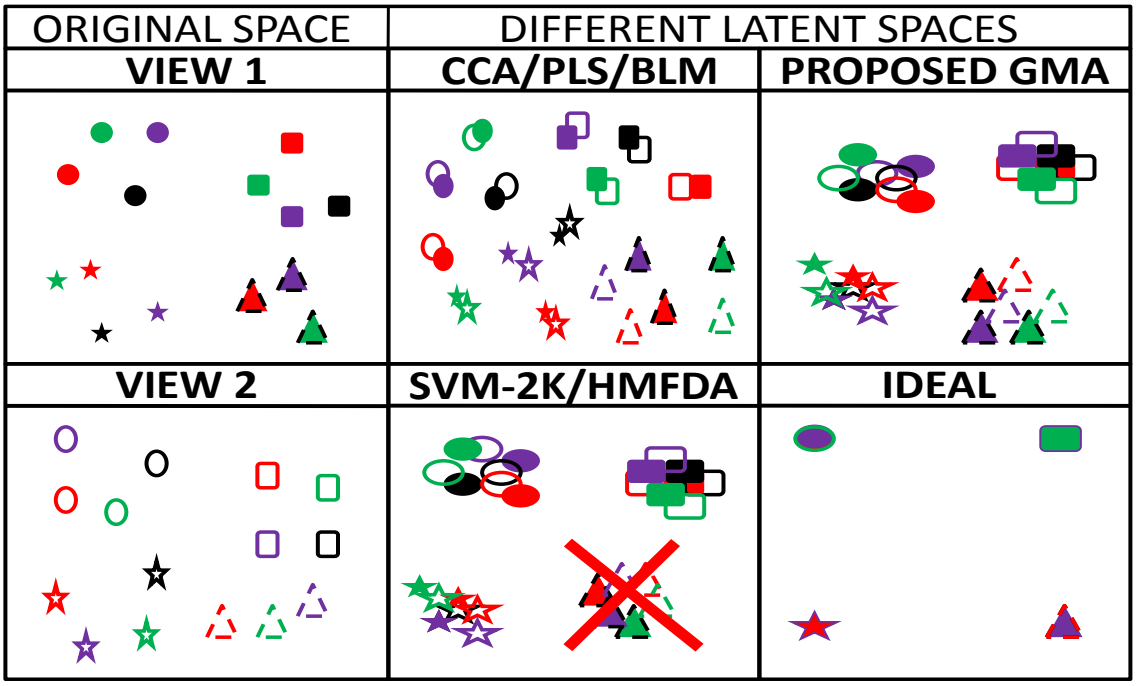


Figure 6.1: A toy example to illustrate the requirements in the common subspace for classification. (Figure best viewed in color)

workhorse for learning a common subspace which is evident from its wide-spread use in vision [39, 107, 109], cross-lingual retrieval[38], cross-media retrieval [51, 95], etc.. Unfortunately, the above mentioned approaches only care about pair-wise closeness in the common subspace so they are not well suited for classification/retrieval. Especially, when within-class variance is large, these methods are bound to perform poorly for classification/retrieval because classification and retrieval both require that within-class samples are united. Moreover, the costly label information that might be available during training is unharnessed. Locality preserving CCA (LPCCA) was introduced to capture the non-linearity present in the data by forcing nearby points in the original feature space to be close in the common subspace as well [124]. However, they did not use the label information and we will see that it is a special instance of our general model. Discriminative CCA (DCCA) uses multi-dimensional labels as the second view, which is just single view scenario with multidimensional labels [123]. CCA is used to match sets of images by maximizing within-set correlation and minimizing between set correlation, which is again a single view scenario with set membership information [55]. We are interested in scenarios in which the data has two different views, along with label information.

A number of supervised approaches to multi-view analysis have also been proposed. Multi-view Fisher Discriminant Analysis (MFDA) learns classifiers in different views by maximizing the agreement between the predicted labels of these classifiers[25]. But, MFDA can only be used for two-class problems. To cope with this, [20] extended MFDA to a multi-class scenario using a Hierarchical clustering approach. In [29], the authors obtained a multi-view version of SVM by constraining the one-dimensional outputs of individual SVM's to be equal. These approaches however, use multi-view data to learn classifiers in each view that are better than the classifiers learned using single-view data only. With

some non-trivial adaptation they can be used for cross-view classification and retrieval, but originally the authors have used them as single-view classifiers trained with multi-view data. The prime objective of this paper is cross-view classification and retrieval. Most importantly, none of MFDA, SVM-2K or HMFDA can classify samples from unseen classes, which is required in many real-world applications such as face recognition, cross-view retrieval and domain adaptation. For example, practical face recognition often requires a classifier that can compare images of unseen subjects (not used in training) at testing time, while cross-view retrieval also requires retrieval of unseen categories.

Finally, some *domain-specific* approaches use domain information to learn discriminative cross-view classifiers. Lighting invariant features are used in [65]. Synthetic virtual images in new pose and lighting conditions are used to train LDA for pose and lighting invariant face recognition in [107]. Geometry assisted hashing is used to counter pose and lighting change in [141]. Use of logistic regression with topic modeling features to obtain semantically meaningful features is used in [95] to extract text and image features for cross-media retrieval. Unfortunately, it might not work for unseen classes or when topic modeling is not effective e.g. face recognition. These approaches are customized to a particular task and such domain information may not be available in general.

Based on the above discussion we conclude that an ideal cross-view classification approach *must* be

- **Supervised(S)**: Use label information for class based discrimination.
- **Generalizable (G)**: Able to analyze new classes that are not used during training.
- **Multi-view (MV)**: Applicable to cross-view classification and retrieval, rather than just using multi-view data for learning.
- **Efficient (E)**: Have an efficiently computed optimal solution.
- **Kernelizable (K)**: Have a kernel extension to model non-linearities.
- **Domain-Independent(DI)**: Applicable to general problems.

Table 6.1 lists some popular approaches and we can see that none of the previous approaches has all the desired properties but the proposed approach has all of them.

6.3 Proposed Approach

Our approach is motivated by the fact that popular supervised and unsupervised feature extraction techniques can be cast as a special form of a quadratically constrained quadratic program (QCQP). Specifically, the optimal projection direction $\hat{\mathbf{v}}$ can be obtained as

$$\begin{aligned} \hat{\mathbf{v}} &= \underset{\mathbf{v}\neq 0}{\operatorname{argmax}}\, \mathbf{v}^T A \mathbf{v} \\ & s.t.\ \mathbf{v}^T B \mathbf{v} = 1 \ or\ \mathbf{v}^T \mathbf{v} = 1 \end{aligned} \tag{6.1}$$

Here, A is some symmetric square matrix and B is a square symmetric Definite Matrix i.e. no eigenvalue of B is equal to 0. Methods that fit this equation include PCA [82, 138], LDA [8, 138], LPP [43, 138], CCA, and MFA [138]. So, we first extend Eqn. 6.1 to a multi-view scenario and then use it with different (A, B) combinations to obtain different common

Table 6.1: Properties of popular approaches for classification and feature extraction. Note that only the proposed GMA approach has all the required properties. **S**: **S**upervised, **G**: **G**eneralizable, **MV**: **M**ulti-**V**iew, **E**: Efficient, **K**: **K**ernelizable, **DI**: **D**omain-**I**ndependent (X indicates presence of property).

Method	**Properties**					
	S	**G**	**MV**	**E**	**K**	**DI**
PCA [82]		X		X	X	X
LDA [8]	X	X		X	X	X
MFA [138]	X	X		X	X	X
LPP [43]	X	X		X	X	X
BLM [128]		X	X	X	X	X
CCA [39]		X	X	X	X	X
PLS [39, 99]		X	X	X	X	X
SVM-2K [29]	X			X	X	X
MFDA [25]	X			X	X	X
HMFDA [20]	X			X	X	X
LPCCA [124]		X	X	X	X	X
DCCA [123]	X	X		X	X	X
SetCCA [55]	X			X	X	X
GMA	**X**	**X**	**X**	**X**	**X**	**X**

subspaces with desired properties. For ease of understanding, we derive the results for two views and later extend it to multiple views.

6.3.1 Generalized Multiview Analysis

We now present a generalization of this framework to a multi-view setting. We first extend Eqn. 6.1 to a multi-view setting in Eqn. 6.2, combining two optimization problems without yet coupling them. Then, in Eqn. 6.6 we constrain the samples from the same content to project to similar locations in the latent space.

A joint optimization of two objective functions over two different vector spaces can be written as

$$\begin{aligned}[\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2] = \underset{\mathbf{v}_1,\mathbf{v}_2}{\operatorname{argmax}} \ & \mathbf{v}_1^T A_1 \mathbf{v}_1 + \mu \mathbf{v}_2^T A_2 \mathbf{v}_2 \\ & s.t.\ \mathbf{v}_1^T B_1 \mathbf{v}_1 = \mathbf{v}_2^T B_2 \mathbf{v}_2 = 1\end{aligned} \tag{6.2}$$

The positive term μ is to bring a balance between the two objectives, because if $\max \mathbf{v}_1^T A_1 \mathbf{v}_1 \gg \max \mathbf{v}_2^T A_2 \mathbf{v}_2$, the joint objective will be biased towards optimizing $\mathbf{v}_1$ and vice-versa. Please note that the optimization problem from Eqn. 6.2 can be solved as a generalized eigen-value problem to obtain $\mathbf{v}_1$ and $\mathbf{v}_2$, but in order to facilitate a stream-lined flow of building up the multi-view extension we couple the constraints with $\gamma = \frac{tr(B_1)}{tr(B_2)}$ to obtain a relaxed version of the problem with a single constraint as

$$\begin{aligned}[\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2] = \underset{\mathbf{v}_1,\mathbf{v}_2}{\operatorname{argmax}} \ & \mathbf{v}_1^T A_1 \mathbf{v}_1 + \mu \mathbf{v}_2^T A_2 \mathbf{v}_2 \\ & s.t.\ \mathbf{v}_1^T B_1 \mathbf{v}_1 + \gamma \mathbf{v}_2^T B_2 \mathbf{v}_2 = 1\end{aligned} \tag{6.3}$$

When $\hat{\mathbf{v}}_1^T B_1 \hat{\mathbf{v}}_1 = \hat{\mathbf{v}}_2^T B_2 \hat{\mathbf{v}}_2$, the constraints in Eqn. 6.2 and Eqn. 6.3 are equivalent. When $\hat{\mathbf{v}}_1^T B_1 \hat{\mathbf{v}}_1 \neq \hat{\mathbf{v}}_2^T B_2 \hat{\mathbf{v}}_2$, the constraint in Eqn. 6.3 is an approximation of the constraints in Eqn. 6.2. We empirically observed that parameter γ did not have much effect on overall performance.

Intuitively, the resulting problem in Eqn. 6.3 is solving the relaxed version of the original optimization problem in two different vector spaces (views). To facilitate understanding, let's consider a multi-view extension of LDA. In this case, $A_i = S_{bi}$, $B_i = S_{wi}$ for $i = 1, 2$ where S_{bi} and S_{wi} are between and within class scatter matrices and $\mathbf{v_1}$ and $\mathbf{v_2}$ are the projection directions in view 1 and 2 respectively. Eqn. 6.3 is jointly solving for LDA projection directions $\hat{\mathbf{v}}_\mathbf{1}$ and $\hat{\mathbf{v}}_\mathbf{2}$ to maximize between class separation and minimize within class variation in each view.

Now we introduce a constraint to couple these projection directions. For cross-view classification we require that the projections (a_1^i and a_2^i) of the exemplars (z_1^i and z_2^i) of the i^{th} content in different views should be close to each other in the projected latent space. a_1^i and a_2^i are defined as

$$a_1^i = \mathbf{v}_1^T \mathbf{z}_1^i \qquad and \qquad a_2^i = \mathbf{v}_2^T \mathbf{z}_2^i \tag{6.4}$$

We chose to maximize covariance between the exemplars from different views to obtain directions to achieve closeness between multi-view samples of the same class. This leads to a closed form solution and better preserves the between class variation as argued in [109]

$$[\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2] = \underset{\mathbf{v}_1, \mathbf{v}_2}{\operatorname{argmax}} \mathbf{v}_1^T Z_1 Z_2^T \mathbf{v}_2 \tag{6.5}$$

Here, Z_i's are the matrices constructed such that i^{th} column in both Z_1 and Z_2 contains exemplars corresponding to the same content. The exemplars can be chosen to suit the problem and feature extraction techniques. For instance, LDA represents a class as the mean of class samples, so class mean can be used as the exemplar.

Without any constraints on $\mathbf{v}_1$ and $\mathbf{v}_2$ the objective in Eqn. 6.5 can be increased indefinitely. But we couple this objective with the constrained objective of Eqn. 6.3 to get the final constrained objective

$$\begin{aligned} [\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2] = \underset{\mathbf{v}_1, \mathbf{v}_2}{\operatorname{argmax}} \; & \mathbf{v}_1^T A_1 \mathbf{v}_1 + \mu \mathbf{v}_2^T A_2 \mathbf{v}_2 + 2\alpha \mathbf{v}_1^T Z_1 Z_2^T \mathbf{v}_2 \\ s.t. \; & \mathbf{v}_1^T B_1 \mathbf{v}_1 + \gamma \mathbf{v}_2^T B_2 \mathbf{v}_2 = 1 \end{aligned} \tag{6.6}$$

Projection directions $\mathbf{v}_1$ and $\mathbf{v}_2$ will tend to balance the original feature extraction optimization with latent space covariance between exemplars that represent the same content. The vector form of Eqn. 6.6 is

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{v}}_1 \\ \hat{\mathbf{v}}_2 \end{bmatrix} = & \underset{\mathbf{v}_1, \mathbf{v}_2}{\operatorname{argmax}} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}^T \begin{bmatrix} A_1 & \alpha Z_1 Z_2^T \\ \alpha Z_2 Z_1^T & \mu A_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} \\ s.t. \; & \begin{bmatrix} \mathbf{v}_1^T & \mathbf{v}_2^T \end{bmatrix} \begin{bmatrix} B_1 & 0 \\ 0 & \gamma B_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = 1 \end{aligned} \tag{6.7}$$

Equivalently,

$$\begin{aligned} \hat{\mathbf{v}} = & \underset{\mathbf{v}}{\text{argmax}} \ \mathbf{v}^T \tilde{A} \mathbf{v} \\ & s.t. \ \mathbf{v}^T \tilde{B} \mathbf{v} = 1 \\ \Rightarrow \quad & \tilde{A}\hat{\mathbf{v}} = \lambda \tilde{B} \hat{\mathbf{v}} \end{aligned} \tag{6.8}$$

Here, $\hat{\mathbf{v}}^T = [\hat{\mathbf{v}}_1^T \ \hat{\mathbf{v}}_2^T]$ and matrices $\tilde{A}$ and $\tilde{B}$ are the square symmetric matrices in Eqn. 6.7.

The final objective function is a standard generalized eigenvalue problem that can be solved using any eigen-solver. It will produce real eigenvectors and eigenvalues because both $\tilde{A}$ and $\tilde{B}$ are square symmetric matrices. When data dimensions are greater than the number of classes, $\tilde{B}$ could be positive semi-definite and the problem becomes ill-posed. We can add a regularizer to the $\tilde{B}$ or project the original feature vectors to a lower dimensional subspace to handle this.

6.3.2 Multiview Extensions

There are several unsupervised and supervised feature extraction techniques with different properties in a single view scenario such as PCA [82], LDA [8], LPP [43], NPE [42], MFA [138] and their kernel versions. On the multi-view side we are already familiar with the most popular unsupervised feature extraction techniques, namely CCA, BLM and PLS. We showed in the last subsection that a feature extraction technique in the form of a QCQP (Eqn. 6.1) can be extended to a multi-view scenario using our framework. Plugging in different (A, B) pairs for different feature extraction techniques in our framework we can obtain multi-view extensions of PCA [82], LDA [8], LPP [43], NPE [42] and MFA [138]. We also show the relation between CCA, BLM and PLS and Generalized Multiview PCA or GMPCA as specific instances of our general framework. For further discussion, we use X_i to denote the *data matrix* with columns that are data samples in view i with the mean subtracted.

6.3.2.1 CCA, BLM, PLS and GMPCA

PCA in the i^{th} view is the following eigen-value problem

$$X_i W_i X_i^T \mathbf{v}_i = \lambda \mathbf{v}_i \tag{6.9}$$

$W_i = I_i/N_i$ with N_i equal to number of samples and I_i is the identity matrix in the i^{th} view. With different A_i, B_i and Z_i's in Eqn. 6.7 we get

- **GMPCA** $A_i = X_i W_i X_i^T$, $B_i = I$, $Z_i = X_i$
- **CCA** $A_i = 0$, $B_i = X_i W_i X_i^T$ and $Z_i = X_i$.
- **BLM** $A_i = X_i W_i X_i^T$, $B_i = I$ and $Z_i = X_i$ i.e. *same as GMPCA*.
- **PLS** $A_i = 0$, $B_i = I$ and $Z_i = X_i$. The difference from our approach is that in PLS eigen-vectors are found using asymmetric deflation of X_i's [39].

So, we see that all four approaches are related to each other under the proposed GMA framework.

6.3.2.2 Generalized Multiview LDA or GMLDA

LDA in the i^{th} view is the following eigenvalue problem

$$X_i W_i X_i^T \mathbf{v}_i = \lambda X_i D_i X_i^T \mathbf{v}_i \tag{6.10}$$

W_i and D_i are $N_i \times N_i$ matrices with $W_i^{kl} = 1/N_i^c$ if X_i^k and X_i^l belong to class c, 0 otherwise, N_i^c is the number of samples for class c in view i and $D_i = I - W_i$ [138, 43]. So, $A_i = X_i W_i X_i^T$, $B_i = X_i D_i X_i^T$ in Eqn. 6.7. For Z_i we have different choices; we can align corresponding samples giving $Z_i = X_i$, or class means, giving $Z_i = M_i$, with M_i defined as the matrix with columns that are class means. We choose class mean as exemplars because LDA tries to collapse all the class samples to the class mean. So if we align class means in different views we expect the samples to be aligned. Under some situations the within-class variation may not be a unimodal Gaussians. In such cases, samples from the same class can be clustered, and the class can be represented by the cluster centers as exemplars.

6.3.2.3 Generalized Multiview Marginal Fisher Analysis

LDA assumes a Gaussian class distribution, a condition that is often violated in real-world problems. Marginal Fisher Analysis, or MFA, is a technique that does not make this assumption, and instead tries to separate different- and compress same-class samples in the feature space [138]. It leads to following eigenvalue problem

$$X_i(S_{bi} - W_{bi})X_i^T \mathbf{v} = \lambda X_i(S_{wi} - W_{wi})X_i^T \mathbf{v} \tag{6.11}$$

here, $S_{(b/w)i}^{kk} = \sum_{kl,k\neq l} W_{(b/w)i}^{kl}$. The *within class compression* or *intrinsic* graph for the i^{th} view is defined as

$$W_{wi}^{kl} = \begin{cases} 1 & : k \in R_i^{k1}(l) \;\; or \;\; l \in R_i^{k1}(k) \\ 0 & : \text{otherwise} \end{cases} \tag{6.12}$$

Here, $R_i^{k1}(l)$ indicates the index set of the $k1$ nearest neighbors of the sample x_i^l in the same class. The *between class separation* or *penalty* graph for i^{th} view is defined as

$$W_{bi}^{kl} = \begin{cases} 1 & : (k,l) \in P_i^{k2}(c_l) \;\; or \;\; (k,l) \in P_i^{k2}(c_k) \\ 0 & : \text{otherwise} \end{cases} \tag{6.13}$$

Here, $P_i^{k2}(l)$ is a set of data pairs that are the $k2$ nearest pairs among the set $\{(k,l) : k \text{ and } l \text{ are not in the same class}\}$. Hence, $A_i = X_i(S_{bi} - W_{bi})X_i^T$, $B_i = X_i(S_{wi} - W_{wi})X_i^T$ and $Z_i = X_i$. Similarly, multi-view extensions of LPP [43](the same as LPCCA [124]) and NPE [42] can be derived.

6.3.3 Kernel GMA

Kernel GMA involves mapping to a non-linear space and then carrying out GMA in that mapped space to obtain projection directions ν_i for the i^{th} view. So, we replace X_i with $\Phi_i = [\phi(x_i^1), \phi(x_i^2) \ldots \phi(x_i^{N_i})]$ and observe that $\nu_i = \Phi_i \tau_i$. The exemplars in kernel space are the columns of $N_i \times z$ matrix $\mathcal{Z}_i = \Phi_i G_i$, N_i = # samples in view i), z (same for each view) is the number of exemplars in each view, and G_i is an appropriately chosen

$N_i \times z$ matrix. For example - G_i is the $N_i \times N_i$ identity matrix if all the samples are chosen to be exemplars and $N_i \times C$ matrix with $G_i^{r,c} = 1/N_i^c$ if the r^{th} sample belongs to class c, $C =$ # of classes and $N_i^c =$ # of samples in class c. The resulting eigenvalue problem $\tilde{\mathcal{A}}\tau = \lambda\tilde{\mathcal{B}}\tau$ will give $N = \sum_{i=1}^{V} N_i$ dimensional eigenvectors τ, which can be broken down into V parts to obtain the dual form of the eigenvectors for V views. These dual vectors will be used to project test sample $\mathbf{t}_i^j$ into the common non-linear latent space as

$$\mathbf{t}_{common}^j = \sum_{n=1}^{N_i} \varphi(\mathbf{t}_i^j.\mathbf{x}_i^n).\tau_i^n = \tau_i^T \phi(\mathbf{t}_i^j) \tag{6.14}$$

Here, $\phi(\mathbf{t}_i^j)$ is an $N_i \times 1$ vector of kernel evaluations of $\mathbf{t}_i^j$ with all the data samples in the i^{th} view.

6.3.4 More than two views

For more than two views simple algebra tells that we need to set $\tilde{A}$ and $\tilde{B}$ as

$$\tilde{A} = \begin{bmatrix} A_1 & \lambda_{12}Z_1Z_2^T & \cdots & \lambda_{1n}Z_1Z_n^T \\ \lambda_{12}Z_1^TZ_2 & \mu_2A_2 & \cdots & \lambda_{2n}Z_2Z_n^T \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{1n}Z_n^TZ_1 & \lambda_{2n}Z_n^TZ_2 & \cdots & \mu_nA_n \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & \gamma_2B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_nB_n \end{bmatrix} \tag{6.15}$$

6.4 Experimental Results

In this section we test the proposed GMA approach on problems for cross-view classification with available class labels, showing improvement over other approaches.

6.4.1 Pose and Lighting Invariant Face Recognition

This is a problem with simultaneous cross view (pose) and within-class (lighting) variation. We use the MultiPIE [37] face dataset, which has 337 subjects' face images taken across 15 different poses, 20 illuminations, 6 expressions and 4 different sessions. We have done experiments using 5 poses ranging from frontal to profile (75°) at an interval of 15°. We have considered 18 lighting conditions for our experiments (illuminations 1 to 18). All the images are cropped (40 by 40 pixels) and aligned using 4 hand annotated fiducial points (eyes, nose tip and mouth) and affine transformations.

In the training phase, multiple images of a person (under different lighting conditions) in two different poses $p1$ and $p2$ are used to learn pairs of pose-specific projection directions $\hat{\mathbf{v}}_\mathbf{1}$ and $\hat{\mathbf{v}}_\mathbf{2}$, respectively. At testing time, gallery and probe images are projected using learned pairs of pose-specific projection directions i.e. a face image in pose p is projected on $\hat{\mathbf{v}}_p$. 1-NN matching is done in the feature space using the normalized correlation score as a metric. We use two different modes for our recognition experiments. **Mode-1** matches the conditions in a number of prior experiments and **Mode-2** highlights our ability to generalize to unseen classes that were not used to obtain the latent space projection directions. In all our experiments, the gallery consists of a single image per individual, taken in the frontal pose with a frontal light (illum 7); probe images come from all poses and illuminations.

- **Mode-1** We use training images of 129 subjects from session 01 (these 129 subjects were selected because they appear in all 4 sessions which allows future evaluation across sessions 03 and 04) under 5 lightings (1, 4, 7, 12 and 17) and testing images of the same subjects from session 02 under all 18 lightings.

- **Mode-2** Training images of 120 subjects from session 01 (different than the one chosen in Mode-1) under 5 lightings (1, 4, 7, 12 and 17); testing images are the same as Mode-1 testing images.

We have used LDA and MFA with the proposed GMA approach and called the resulting approach GMMFA and GMLDA respectively. A naive way to obtain discriminant directions in two views is to learn a common subspace using CCA followed by LDA in the latent space (CCA+LDA) or LDA in individual spaces followed by CCA to get a common space (LDA+CCA). Surprisingly, neither of these approaches has been used before and we found that even these naive approaches outperform some competitive approaches. LDA, PCA, CCA, BLM, CCA+LDA and LDA+CCA are implemented by us. PLS, BLM and CCA have been used before for pose invariant face recognition to achieve state-of-the-art results on the CMU PIE dataset using PLS (code[1]) [109]. However, we find that with simultaneous pose and lighting variations all three perform poorly. Performance for Gabor [65], Local Feature Hashing or LFH [141], PittPatt [141], Sparse coding [133] are taken directly from the papers. Since, all the implemented approaches lead to large eigenvalue problems, we use PCA to reduce the data dimension before feeding it to any of the feature extraction techniques. We kept the top p principal components that retained 95% of the variance. For GMA based approaches we fix $\alpha = 10$, $\mu = 1$, $\gamma = \frac{tr(B_1)}{tr(B_2)}$, $k1 = 50$, $k2 = 400$ (for GMMFA), and all samples are taken as exemplars for both GMMFA and GMLDA. Parameters for MFA ($k1$ and $k2$) were selected based on the guidelines given in [138]. For simple LDA and PCA, different illumination images in gallery and probe poses are used together to learn common projection directions. The dimension of the feature space is selected by choosing the top k eigenvectors that contain 98% of the total eigenvalues produced by the eigenvalue problems involved in finding projection directions. We tried similar approaches to automatically determine the dimension for PLS based classification but the results were very poor. So for PLS only, we did testing for all possible dimensions and report the best accuracy. While reporting the results from [65] we have considered results for the selected 18 illumination conditions only. PittPatt is a commercial face recognition software and its results were taken directly from [141]. LFH uses a hashing technique with SIFT features for face recognition and frontal, 45° and 90° in the gallery for pose robustness in contrast to our approach in which we have used only frontal pose in the gallery. Use of SIFT features provides some tolerance to pose, and a multi-pose gallery makes matching possible across different poses. The results for LFH and PittPatt are reported using the same 129 subjects from session 02 used in our testing set with gallery images in the left illumination condition, whereas, we have used frontal illumination as the gallery image. However, we found that using any of the 18 illuminations as gallery with GMLDA and GMMFA resulted in negligible differences in performance compared to those reported in Table 6.2. In [133], the authors have used a sparse representation for simultaneous registration and recognition. They have reported results for pose and lighting invariant face recognition for 15° probe pose only, under all illuminations with a gallery of 249 subjects and reported 77.5% accuracy whereas we have used a gallery of

[1] http://www.cs.umd.edu/~djacobs/pubs_files/PLS_Bases.m

Table 6.2: Performance for MultiPIE pose and lighting invariant face recognition in **Mode-1**. Some approaches from other published works have not reported results for all pose differences; the absence is indicated by '-'.

Method	Probe pose					**Avg**
	15°	30°	45°	60°	75°	
PCA	15.3	5.3	6.5	3.6	2.6	6.7
PLS [109]	39.3	40.5	41.6	41.1	38.7	40.2
BLM [109]	46.5	55.1	59.9	63.6	61.8	57.4
CCA [109]	92.1	89.7	88.0	86.1	83.0	83.5
LDA[a]	98.0	94.2	91.7	84.9	79.0	89.5
CCA+LDA	96.4	96.0	93.6	86.2	83.6	91.2
LDA+CCA	95.9	94.9	93.6	91.3	89.9	93.1
PittPatt [141] [a]	94	34.0	3.0	–	–	–
LFH [141][a]	63	58	61	41	43	53.2
Sparse [133][a]	77.5	–	–	–	–	–
GMLDA	**99.7**	**99.2**	**98.6**	**94.9**	**95.4**	**97.6**
GMMFA	**99.7**	**99.0**	**98.5**	**95.0**	**95.5**	**97.5**

[a] Domain-dependent for cross-view classification

129 subject and report 99.7%.

The results from the experiments done under Mode-2 are shown in Table 6.3. It is clear that GMMFA and GMLDA outperformed other approaches except [65] for large pose differences but overall performance of the proposed GMA based approach is better than all the domain-specific as well as generic approaches. Surprisingly, LDA performance is better than CCA, which is not expected due to the large pose difference. This unexpected observation indicates the importance of using label information in training. It also explains the improvements offered by GMLDA, because GMLDA is a fusion of CCA and LDA. Unfortunately, LDA cannot be used in cases when the data dimensions are different in different views, for example - image-text or text-link cases.

6.4.2 Text-Image Retrieval

Text-image retrieval is yet another cross-view problem that requires a common representation. We show results on two publicly available datasets - Pascal VOC 2007 [51, 50, 27] and Wiki Text-Image data [95]. Pascal data consists of 5011/4952(training/testing) image-tag pairs collected by the authors in [51, 50, 27] and it has 20 different classes. We used the publicly available features [2] consisting of histograms of bag-of-visual-words, GIST and color for images and *relative* and *absolute* tag ranks for text with a Chi-square kernel (see [51] for details). Some images are multi-labeled so we selected images with only one object from the training and testing set, which resulted in 2808 training and 2841 testing data. The category of the object is used as the content so we have a 20 class problem. A second data set, Wiki Text-image, consists of 2173/693(training/testing) image-text pairs with 10 different classes. We have used the same data as supplied by the authors[3]. It has a 10 dimensional latent Dirichlet allocation model [13]

[2] http://www.cs.utexas.edu/~grauman/research/datasets.html

[3] http://www.svcl.ucsd.edu/projects/crossmodal/

Table 6.3: Performance for MultiPIE pose and lighting invariant face recognition in **Mode-2**. Some approaches from other published works have not reported results for all pose differences; the absence is indicated by '-'.

Method	Probe pose					**Avg**
	15°	30°	45°	60°	75°	
PCA	14.0	4.9	6.1	3.3	2.4	6.2
PLS [109]	29.0	26.2	23.3	17.3	12.4	21.6
BLM [109]	53.9	44.6	34.3	22.5	20.8	35.3
CCA [109]	79.5	62.2	46.1	19.5	14.4	44.3
LDA[a]	88.5	68.9	56.2	21.7	21.0	51.3
CCA+LDA	79.5	58.0	44.6	21.0	20.1	44.6
LDA+CCA	74.9	54.7	37.8	13.4	11.0	38.4
Gabor [65][a]	77.9	74.5	58.1	**45.2**	**31.0**	57.4
GMLDA	**92.6**	**80.9**	**64.4**	32.3	28.4	**59.7**
GMMFA	**92.7**	**81.1**	**64.7**	32.6	28.6	**59.9**

[a] Domain-dependent for cross-view classification

Table 6.4: mAP scores for image and text query on Wiki text-image data.

Query	Others					Proposed	
	PLS	BLM	CCA	SM	SCM	GMMFA	GMLDA
Image	0.207	0.237	0.182	0.225	**0.277**	0.264	0.272
Text	0.192	0.144	0.209	0.223	0.226	0.231	**0.232**
Average	0.199	0.191	0.196	0.224	0.252	0.248	**0.253**

Table 6.5: mAP scores on Pascal data.

Query	Others		Proposed	
	KPLS	KCCA	KGMMFA	KGMLDA
Image	0.279	0.298	0.421	**0.427**
Text	0.232	0.269	0.328	**0.339**
Average	0.256	0.283	0.375	**0.383**

based text features and 128 dimensional SIFT histogram image features (see [95] for more details). Both data sets have class labels that can be leveraged in our proposed GMA framework to achieve within-class invariance. The task is to retrieve images/text from a database for a given query text/image. A correct retrieval is one that belongs to the same class as the query. So we want more and more correct matches in the top k documents for a better retrieval.

Semantic Correlation Matching (SCM) with a linear kernel [95] has shown state-of-the-art performance for Wiki data, so we have compared the proposed GMLDA and GMMFA with CCA, PLS, BLM, SCM and Semantic Matching (SM) [95]. SM corresponds to using Logistic regression in the image and text feature space to extract semantically similar feature to facilitate better matching. SCM refers to the use of Logistic regression in the space of CCA projected coefficients (a two-stage learning process). Results for SM and SCM are directly taken from the paper [95]. The authors in [51] have shown the

advantage of using a Chi-square kernel over a linear mapping so we have used a Chi-square kernel for Pascal data for all the methods resulting in KernelCCA (KCCA), KernelPLS (KPLS), KernelGMLDA (KGMLDA) and KernelGMMFA (KGMMFA). For GMA based approaches we fix $\alpha = 100$, $\mu = 1$, $\gamma = \frac{tr(B_1)}{tr(B_2)}$, $k1 = 500$, $k2 = 2200$ (for GMMFA) and all the samples belonging to a class are taken as exemplars for both GMMFA and GMLDA. We have kept same number of dimensions for all the methods as mentioned in [95] and [51] i.e. 10 for Wiki and 20 for Pascal. Precision at 11 different recall levels {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} is used to evaluate the performance. The Mean Average Precision (mAP) scores= $\frac{1}{11}\sum_{r=0}^{r=1} Precision_r$ for text and image query for Wiki and Pascal data are listed in Table3 and Table4, respectively. It is evident that GMLDA and GMMFA outperform CCA, PLS, BLM and SM on Wiki data. Surprisingly, our generic single-stage approach's performance is similar to the domain specific two-stage SCM approach. We also outperformed KCCA and KPLS on Pascal data. The improvement is more for Pascal data because there are more classes (20 vs 10) and more testing samples (2841 vs 693) as compared to Wiki data, which requires better union of within-class samples for better performance.

6.5 Conclusion

We have proposed a novel generic framework for multi-view feature extraction by extending several unsupervised and supervised feature extraction techniques to their multi-view counterpart. We call the proposed framework Generalized Multiview Analysis or GMA. It is a first step towards unified multi-view feature extraction. The proposed approach is general and kernelizable, simultaneously learns multi-view projection directions and generalizes across unseen classes. We have shown that any feature extraction technique in the form of a generalized eigenvalue problem can be extended to its multi-view counterpart and we have used GMA to obtain multi-view counterparts of PCA, LDA, LPP, NPE and MFA. We have also unified CCA, PLS, BLM as specific instances of Generalized Multiview PCA. Using LDA and MFA in our framework we have significantly outperformed all generic and most of the domain specific approaches for pose and lighting invariant face recognition. Using the same general framework we have also shown state-of-the-art results on text-image retrieval on Wiki data and outperformed generic approaches on Pascal data. GMA has outperformed CCA for all tasks when label information is available therefore, proving to be a superior alternative for CCA under similar conditions.

Chapter 7

Concluding Remarks and Future Directions

This dissertation studies the problem where data can occur in multiple modalities and it is required to extract and match task-dependent content across modalities. We work with the postulate that humans extract and store task-dependent content from multiple modalities as a *common representation* that affords seamless cross-modal content matching. The primary contribution of this dissertation lies in developing novel algorithms for visual content extraction and cross-modal content matching. These algorithms can be used while dealing with pose-invariant face recognition, text-image matching and multi-modal biometrics. Some specific contributions of this dissertation are -

1. Development of a novel recursive neural network for semantic segmentation of images, termed Recursive Context Propagation Network or RCPN [112, 111].

2. Modeling pose-invariant face recognition, sketch-photo recognition and low resolution vs high resolution face recognition as cross-modal content matching problems and obtaining a Partial Least Square based common representation to achieve impressive results on standard datasets [109]. It was further extended to a two-stage discriminative architecture to handle pose-errors in [108].

3. Development of a novel framework to extend any feature extraction technique, based on generalized eigenvalue analysis, to its multi-view counterpart [110]. The resulting framework is kernelizable and applicable to more than two modalities. It has been used for text-image retrieval and shown to outperform state-of-the-art approaches for both the problems.

7.1 Future Directions

I believe that the common representation hypothesis has a lot of merits and is probably the key to many cross-modal matching problems. However, either pure statistics based or domain dependent solutions are not expected to provide a complete solution. Therefore, we require a mix of these two approaches for acceptable solutions under most conditions. The recent rise and spectacular success of deep neural network based approaches for visual and textual content extraction dictates the use of the learned features from deep nets. Unfortunately, the space of multi-modal matching via deep learning is less explored and only a few general approaches exist [84, 121, 3]. Recently, impressive performance has been demonstrated on sentence generation from images and image retrieval using complex text queries in [54]. However, it still relies on the richness of the content extraction from individual modality that leaves a lot of room for improvement in developing deep algorithms for cross-modal content matching. Moreover, the requirement of large amount of labeled data for training a deep net also puts serious limitations on the versatility of deep net based approaches for different problem domains. Therefore, collecting large amount of annotated data for various tasks coupled with domain-dependent

deep neural architectures seems to be a promising future direction for solving cross-modal content matching problem.

Sometimes it's possible that different modalities have complementary, missing or noisy content and it becomes mandatory to combine the contents extracted from different modalities to achieve the best results. This situation calls for multi-modal content fusion. For example, it would be very difficult even for humans to assemble an office desk without the aid of a 2D diagram just by reading the written instructions and vice-versa. However, having both of them makes the task relatively simpler. This simple looking task of multi-modal content fusion involves some fundamental problems that can create hindrance such as how to deal with the different representation of content, what common representation should be selected for efficient fusion etc. Some crucial applications that can benefit from multi-modal content fusion are: visual object recognition, scene understanding etc. The key idea is to represent the content from different modalities in a form that makes multi-modal content fusion possible. Therefore, formulating domain dependent algorithms for multi-modal content fusion can be a promising future direction.

Relevant Publications

1. Abhishek Sharma, Oncel Tuzel, David W. Jacobs: Deep Hierarchical Parsing for Semantic Segmentation. IEEE CVPR 2015.

2. Abhishek Sharma, Oncel Tuzel, Ming Yu Liu: Recursive Context Propagation Network for Scene Labeling. NIPS 2014.

3. Abhishek Sharma, Murad Al-Haj, Jonghyun Choi, Larry S. Davis and David W. Jacobs, "Robust Pose Invariant Face Recognition using Coupled Latent Space Discriminant Analysis", CVIU 116 (11), 1095-1110, 2012.

4. Abhishek Sharma, Abhishek Kumar, Hal Daume III and David W. Jacobs, "Generalized Multiview Analysis: A Discriminative Latent Space", IEEE CVPR 2012.

5. Abhishek Sharma and David W Jacobs, " Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch", IEEE CVPR 2011.

6. Jonghyun Choi, **Abhishek Sharma**, David W Jacob and Larry S Davis, " Data insufficiency in sketch versus photo face recognition", IEEE CVPR Workshops, 2012.

Bibliography

[1] Partial least square tutorial. *http://www.statsoft.com/textbook/partial-least-squares/#NIPALS*.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, 2012.

[3] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 1247–1255. JMLR Workshop and Conference Proceedings, 2013.

[4] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012.

[5] A. B. Ashraf, S. Lucey, and T. Chen. Learning patch correspondences for improved viewpoint invariant face recognition. *in Proc. IEEE CVPR*, pages 1–8, 2008.

[6] A. Asthana, T. Marks, M. Jones, K. Tieu, and R. MV. Fully automatic pose-invariant face recognition via 3d pose normalization. *in Proc. IEEE ICCV*, pages 937–944, 2011.

[7] J. Baeka and M. Kimb. Face recognition using partial least squares components. *Pattern Recognition*, 37:1303–1306, 2004.

[8] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs.fisherfaces: recognition using class specific linear projection. *IEEE Trans. Patt. Anal. Mach. Intel*, 19:711–720, 1997.

[9] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *in Proc. IEEE CVPR*, 2011.

[10] Y. Bengio, I. J. Goodfellow, and A. Courville. *Deep Learning*. 2014. Book in preparation for MIT Press.

[11] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Patt. Anal. Mach. Intel.*, 25(9):1063–1074, 2003.

[12] M. Blaschko and C. Lampert. Correlational spectral clustering. *in Proc. IEEE CVPR*, pages 1–8, 2008.

[13] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[14] A. Boulesteix and K. Strimmer. Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. *Briefings in Bioinformatics, Advance Access publication*, 8(1):32–44, 2006.

[15] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *in Proc. IEEE CVPR*, 2012.

[16] C. Castillo and D. Jacobs. Using stereo matching with general epipolar geometry for 2-d face recognition across pose. *IEEE Trans. Patt. Anal. Mach. Intel.*, 31(12):2298–2304, 2009.

[17] C. Castillo and D. Jacobs. Wide-baseline stereo for face recognition with large pose variation. *in Proc. IEEE CVPR*, pages 537–544, 2011.

[18] X. Chai, X. C. S. Shan, and W. Gao. Locally linear regression for pose invariant face recognition. *IEEE Tran. Image Processing*, 16(7):1716–1725, 2007.

[19] X. Chai, S. Shan, X. Chen, and W. Gao. Locally linear regression for pose invariant face recognition. *IEE TIP*, 16(7):1716–125, 2007.

[20] Q. Chen and S. Sun. Hierarchical multi-view fisher discriminant analysis. ICONIP '09, pages 289–298, 2009.

[21] Y. L. Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Advances in neural information processing systems 2. chapter Handwritten Digit Recognition with a Back-propagation Network, pages 396–404. 1990.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *in Proceedings of IEEE CVPR*, pages 886–893, 2005.

[23] C. Dhanjal, S. Gunn, and J. Taylor. Efficient sparse kernel feature extraction based on partial least squares. *IEEE Trans. Patt. Anal. Mach. Intel.*, 31(8):1947–1961, 2009.

[24] J. DiCarlo, D. Zoccolan, and N. Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415 – 434, 2012.

[25] T. Diethe, D. Hardoon, and J. Shawe-Taylor. Constructing nonlinear discriminants from multiple data views. In *ECML PKDD*, pages 328–343. Springer-Verlag, 2010.

[26] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.*, 11:625–660, 2010.

[27] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007 (voc2007) results. http://www.pascalnetwork. org/challenges/voc/voc2007/workshop/index.html.

[28] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE TPAMI*, August 2013.

[29] J. D. R. Farquhar, H. Meng, S. Szedmak, D. R. Hardoon, and J. Shawe-taylor. Two view learning: Svm-2k, theory and practice. In *NIPS*. MIT Press, 2006.

[30] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.

[31] R. Fergus and D. Eigen. Nonparametric image parsing using adaptive neighbor sets. *IEEE CVPR*, 2012.

[32] B. Fröhlich, E. Rodner, and J. Denzler. Semantic segmentation with millions of features: Integrating multiple cues in a combined random forest approach. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part I*, pages 218–231, 2013.

[33] C. Goller and A. Kchler. Learning task-dependent distributed representations by backpropagation through structure. *Int Conf. on Neural Network*, 1995.

[34] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. *IEEE ICCV*, 2009.

[35] R. Gross, I. Matthews, and S. Baker. Appearance-based face recognition and light-fields. *IEEE Trans. Patt. Anal. Mach. Intel.*, 26(4):449–465, 2004.

[36] R. Gross, I. Matthews, and S. Baker. Appearance-based face recognition and light-fields. *TPAMI*, 26(4):449–465, 2004.

[37] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multipie. *Image and Vision Computing*, 28(5):807–813, 2010.

[38] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: an overview with application to learning methods. *Neural Comput*, 16:2639–2664, 2004.

[39] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16:2639–2664, 2004.

[40] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011.

[41] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision (ECCV)*, 2014.

[42] X. He, D. Cai, S. Yan, and H. Zhang. Neighborhood preserving embedding. In *ICCV*, volume 2, pages 1208 –1213, 2005.

[43] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. *IEEE TPAMI*, 27(3):328–340, 2005.

[44] X. He, R. S. Zemel, and M. A. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *in Proc. of IEEE CVPR*, pages 695–703, 2004.

[45] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, pages 30–43, 2008.

[46] P. B. H.F. Chen and D. Jacobs. In search of illumination invariance. *in Proc. IEEE CVPR*, pages 254–261, 2010.

[47] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[48] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *Int. J. Comput. Vision*, 75(1):151–172, 2007.

[49] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Netw.*, 4(2):251–257, 1991.

[50] S. Hwang and K. Grauman. Reading between the lines: Object localization using implicit cues from image tags. *TPAMI, IEEE*, 2011.

[51] S. J. Hwang and K. Grauman. Accounting for the relative importance of objects in image retrieval. In *BMVC*, pages 1–12, 2010.

[52] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[53] T. Kanade and A. Yamada. Multi-subregion based probabilistic approach toward pose-invariant face recognition. *Proc. IEEE Int. Symposium Robotics Automation*, 2:954–959, 2003.

[54] A. Karpathy and F.-F. Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE CVPR*, 2015.

[55] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *TPAMI, IEEE*, 29(6):1005 –1018, 2007.

[56] B. Klare, Z. Li, and A. Jain. Matching forensic sketches to mug shot photos. *TPAMI*, 33(3):639–646, 2011.

[57] B. Klare, Z. Li, and A. K. Jain. Matching forensic sketches to mugshot photos. *TPAMI*, 33(3):639–646, 2011.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[59] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. *IEEE CVPR*, 2010.

[60] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr. What, where and how many? combining object detectors and crfs. In *in Proc. of ECCV*, pages 424–437, 2010.

[61] L. Ladick, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, 100(2):122–133, 2012.

[62] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*, pages 282–289, 2001.

[63] V. Lempitsky, A. Vedaldi, and A. Zisserman. A pylon model for semantic segmentation. *NIPS*, 2011.

[64] A. Li, X. C. S. Shan, and W. Gao. Maximizing intra-individual correlations for face recognition across pose differences. *in Proc. IEEE CVPR*, pages 605–611, 2009.

[65] A. Li, S. Shan, and W. Gao. Coupled bias-variance trade off for cross-pose face recognition. *IEEE Transaction Image Processing*, 21(1):305–315, 2012.

[66] A. Li, S. Shan, X.Chen, and W. Gao. Cross-pose face recognition based on partial least squares. *Pattern Recognition Letters*, 32(15):1948–1955, 2011.

[67] B. Li, H. Chang, S. Shan, and X. Chen. Aligning coupled manifolds for face hallucination. *IEEE Signal Processing Letters*, 16(11):957–960, 2009.

[68] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, London, UK, UK, 1995.

[69] X. Li, J. Ma, and S. Lia. Novel face recognition method based on a principal component analysis and kernel partial least square. *IEEE ROBIO*, pages 1773–1777, 2007.

[70] C. Liu, H. Y. Shum, and W. T. Freeman. Face hallucination: theory and practice. *IJCV*, 75(1):115–134, 2007.

[71] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE TPAMI*, 33(12), Dec 2011.

[72] D. C. Liu, J. Nocedal, and D. C. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[73] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. *IEEE CVPR*, 2011.

[74] Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma. Nonlinear approach for face sketch synthesis and recognition. *IEEE CVPR*, pages 1005–1010, 2005.

[75] X. Liu and T. Chen. Pose-robust face recognition using geometry assisted probabilistic modeling. *in Proc. IEEE CVPR*, pages 502–509, 2005.

[76] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE CVPR*, 2015.

[77] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[78] S. Lucey and T. Chen. A viewpoint invariant, sparsely registered, patch based, face verifier. *Int. Journal of Computer Vision*, 80:58–71, 2008.

[79] T. Malisiewicz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, June 2008.

[80] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. *IEEE CVPR*, 2014.

[81] R. Mottaghi, S. Fidler, J. Yao, R. Urtasun, and D. Parikh. Analyzing semantic segmentation using hybrid human-machine crfs. *IEEE CVPR*, 2013.

[82] M.Turk and A.Pentland. Eigenfaces for recognition. *Journal Cognitive Neuroscience*, 3(1):71–86, 1991.

[83] D. Munoz, J. A. Bagnell, and M. Hebert. Stacked hierarchical labeling. *ECCV*, 2010.

[84] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning (ICML)*, 2011.

[85] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. In *Progress in Brain Research*, 2006.

[86] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. *CVPR*, 2010.

[87] P. H. O. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene parsing. *ICML*, 2014.

[88] G. Polatkan and O. C. Tuzel. Compressed inference for probabilistic sequential models. *UAI*, pages 609–618, 2011.

[89] J. B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46:77–105, 1990.

[90] U. Prabhu, J. Heo, and M. Savvides. Unconstrained pose invariant face recognition using 3d generic elastic models. *IEEE Trans. Patt. Anal. Mach. Intel.*, 33(10):1952–1961, 2011.

[91] S. Prince, J. W. J.H. Elder, and F. Felisbert. Tied factor analysis for face recognition across large pose differences. *IEEE Trans. Patt. Anal. Mach. Intel.*, 30(6):970–984, 2008.

[92] S. Prince, P. Li, Y. Fu, U. Mohammed, and J. Elder. Probabilistic models for inference about identity. *IEEE Trans. Patt. Anal. Mach. Intel.*, 34(1):144–157, 2012.

[93] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007.

[94] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. *In Proc. of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010.

[95] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM Multimedia*, pages 251–260, 2010.

[96] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. *Proc. ECCV*, 4:3–19, 2002.

[97] S. Romdhani, T. Vetter, and D. J. Kriegman. Face recognition using 3-d models: pose and illumination. *proc. of IEEE*, 94(11):1977–1999, 2006.

[98] R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. *In Subspace, latent structure and feature selection techniques, Lecture Notes in Computer Science, Springer*, pages 34–51, 2006.

[99] R. Rosipal and N. Kramer. Overview and recent advances in partial least squares. *LNCS*, pages 34–51, 2006.

[100] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, 1988.

[101] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth. Stixmantics: A medium-level model for real-time semantic scene understanding. *ECCV*, 2014.

[102] T. Scharwchter, M. Enzweiler, U. Franke, and S. Roth. Efficient multi-cue scene segmentation. In *Pattern Recognition*, volume 8142 of *Lecture Notes in Computer Science*, pages 435–445. 2013.

[103] J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Comput.*, 4(2):234–242, Mar. 1992.

[104] M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, 1997.

[105] W. Schwartz, H. Guo, and L. Davis. A robust and scalable approach to face identification. *in Proc. ECCV*, pages 476–489, 2010.

[106] S. Shan, Y. Chang, W. Gao, B. Cao, and P. Yang. Curse of mis-alignment in face recognition: problem and a novel mis-alignment learning solution. *IEEE conf. Auto Face Gesture Recog.*, pages 314–320, 2004.

[107] A. Sharma, A. Dubey, P. Tripathi, and V. Kumar. Pose invariant virtual classifiers from single training image using novel hybrid-eigenfaces. *Neurocomputing*, 73(10):1868–1880, 2010.

[108] A. Sharma, M. A. Haj, J. Choi, L. S. Davis, and D. W. Jacobs. Robust pose invariant face recognition using coupled latent space discriminant analysis. *Comput. Vis. Image Underst.*, 116(11):1095–1110, Nov. 2012.

[109] A. Sharma and D. W. Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *CVPR*, pages 593–600. IEEE, 2011.

[110] A. Sharma, A. Kumar, H. D. III, and D. W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *IEEE Conf. on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2160–2167, 2012.

[111] A. Sharma, O. Tuzel, and D. W. Jacobs. Deep hierarchical parsing for semantic segmentation. *IEEE CVPR*, 2015.

[112] A. Sharma, O. Tuzel, and M. Y. Liu. Recursive context propagation network for semantic segmentation. *NIPS*, 2014.

[113] J. Shawe-Taylor and N. Christianini. Kernel methods for pattern analysis. *Cambridge University Press*, 2004.

[114] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *IEEE CVPR*, pages 1–8, 2008.

[115] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vision*, 81(1):2–23, 2009.

[116] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Trans. Patt. Anal. Mach. Intel.*, 25(12):1615–1618, 2003.

[117] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[118] G. Singh and J. Kosecka. Nonparametric scene parsing with adaptive feature relevance and semantic context. *IEEE CVPR*, 2013.

[119] R. Socher, C. C.-Y. Lin, A. Y. Ng, and C. D. Manning. Parsing natural scenes and natural language with recursive neural networks. *ICML*, 2011.

[120] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

[121] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems 25*, pages 2222–2230. 2012.

[122] V. Struc and N. Pavesic. Gabor-based kernel partial-least-squares discrimination features for face recognition. *Informatica*, 20(1), 2009.

[123] L. Sun, S. Ji, and J. Ye. Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis. *TPAMI*, 33(1):194 – 200, 2011.

[124] T. Sun and S. Chen. Locality preserving cca with applications to data visualization and pose estimation. *Image and Vision Computing*, 25(5):531 –543, 2007.

[125] D. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Patt. Anal. Mach. Intel.*, 18(8):831–836, 1996.

[126] D. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Patt. Anal. Mach. Intel.*, 18(8):831–836, 1996.

[127] X. Tang and X. Wang. Face sketch recognition. *IEEE Transactions on Circuits Systems for Video Technology*, 14(1):50–57, 2004.

[128] J. Tenenbaum and W. Freeman. Separating style and content with bilinear models. *Neural Comp.*, 12(6):1247–1283, 2000.

[129] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. *IEEE CVPR*, 2013.

[130] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. *IJCV*, 101:329–349, 2013.

[131] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. *IEEE CVPR*, 2003.

[132] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453, 2006.

[133] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, and Y. Ma. Towards a practical face recognition system: Robust registration and illumination by sparse representation. *in Proc. IEEE CVPR*, 42:597–604, 2009.

[134] H. Wang, S. Li, and Y. Wang. Face recognition under varying lighting conditions using self quotient image. *in Proc. IEEE Int. Conf. Auto. Face Gesture Recog.*, pages 819–824, 2004.

[135] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *TPAMI*, 31(11):1955–1967, 2009.

[136] L. Wiskott, J. Fellous, N. Kruger, and C. V. der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Patt. Anal. Mach. Intel.*, 19(7):–, 1997.

[137] B. Xiao, X. Gao, D. Tao, Y. Yuan, and J. Li. Photo-sketch synthesis and recognition based on subspace learning. *Neurocomputing*, 73:840–852, 2010.

[138] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE TPAMI*, 29(1):40–51, 2007.

[139] J. Yang, B. Price, S. Cohen, and M.-H. Yang. Context driven scene parsing with attention to rare classes. *CVPR*, pages 3294–3301, 2014.

[140] J. Yang, H. Tang, Y. Ma, and T. Huang. Face hallucination via sparse coding. *IEEE ICIP*, pages 1264–1267, 2008.

[141] Z. Zeng, T. Fang, S. Shah, and I. Kakadiaris. Local feature hashing for face recognition. In *IEEE BTAS*, pages 1–8, 2009.

[142] Y. Zhuang, J. Zhang, and F. Wu. Hallucinating faces: Lph super-resolution and neighbor reconstruction for residue compensation. *Pattern Recognition*, 40:3178–3194, 2007.