WEAKLY SUPERVISED LEARNING FOR UNCONSTRAINED FACE PROCESSING

A Dissertation Presented

by

GARY B. HUANG

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2012

Computer Science

WEAKLY SUPERVISED LEARNING FOR UNCONSTRAINED FACE PROCESSING

A Dissertation Presented

by

GARY B. HUANG

Approved as to style and content by:

Erik Learned-Miller, Chair

Allen Hanson, Member

Andrew McCallum, Member

John Staudenmayer, Member

Lori A. Clarke, Department Chair
Computer Science

To my parents.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Erik Learned-Miller. From funding me a summer early and getting me to my first publication shortly after, to introducing me to deep learning and supporting me through the aftermath, he has been a constant source of wisdom and good humor. I will miss our discussions of his latest research ideas and our debates on ethics.

I would also like to thank Honglak Lee. His help was instrumental in grasping both the theoretical and practical issues in deep learning, and his dedication to producing top-notch research is an inspiration. I would also like to thank Allen Hanson, Andrew McCallum, and John Staudenmayer for their guidance and many helpful suggestions. An important thanks as well to Daphne Koller, Ben Taskar, and Geremy Heitz, who started me down this path by introducing me to artificial intelligence and research as an undergrad at Stanford.

Many thanks to all my friends in the UMass Vision Lab, including Jerod Weinman, Vidit Jain, Moe Mattar, Andrew Kae, Manjunath Narayana, Adam Williams, Jacqueline Feild, Laura Sevilla Lara, David Smith, Carl Doersch, Yi Ding, Dan Xie, and Benjamin Mears. I will miss the LIVING meetings.

A special thanks to Amherst and its denizens; I could not have asked for a better Walden at which to pursue my studies.

Lastly, to my family: The deepest thanks to Mary, for being there for me through every step on our twin journeys, and to my parents (and little brother), for being all a boy could ever ask for. Without the love and support of all my family, this would not have been possible.

ABSTRACT

WEAKLY SUPERVISED LEARNING FOR UNCONSTRAINED FACE PROCESSING

MAY 2012

GARY B. HUANG

B.Sc., STANFORD UNIVERSITY

M.Sc., STANFORD UNIVERSITY

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Erik Learned-Miller

Machine face recognition has traditionally been studied under the assumption of a carefully controlled image acquisition process. By controlling image acquisition, variation due to factors such as pose, lighting, and background can be either largely eliminated or specifically limited to a study over a discrete number of possibilities. Applications of face recognition have had mixed success when deployed in conditions where the assumption of controlled image acquisition no longer holds. This dissertation focuses on this unconstrained face recognition problem, where face images exhibit the same amount of variability that one would encounter in everyday life.

We formalize unconstrained face recognition as a binary pair matching problem (verification), and present a data set for benchmarking performance on the unconstrained face verification task. We observe that it is comparatively much easier to obtain many examples of unlabeled face images than face images that have been labeled with identity or other higher level information, such as the position of the eyes

and other facial features. We thus focus on improving unconstrained face verification by leveraging the information present in this source of weakly supervised data.

We first show how unlabeled face images can be used to perform unsupervised face alignment, thereby reducing variability in pose and improving verification accuracy. Next, we demonstrate how deep learning can be used to perform unsupervised feature discovery, providing additional image representations that can be combined with representations from standard hand-crafted image descriptors, to further improve recognition performance. Finally, we combine unsupervised feature learning with joint face alignment, leading to an unsupervised alignment system that achieves gains in recognition performance matching that achieved by supervised alignment.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Face processing is an area of research within computer vision that focuses on the automatic machine understanding of human faces, encompassing tasks such as detection of human faces in an image, alignment of the face to a canonical position or localization of facial features (*e.g.* eyes, nose) on the face, and recognition of person identity from a face image. Due to the nature of working specifically with face images, such research has the potential for many real-world applications in areas such as security, biometrics, human-computer interaction, and photo organization and search.

As face processing research has progressed, commercial application has followed, with an early notable example being face detection. In 2001, Viola and Jones developed a real-time system for accurate automatic detection of faces [107]. Beginning in 2005, such technology was introduced into consumer-level digital cameras, and today, is a standard feature on most digital cameras, used to assist in properly setting parameters such as focus, exposure, and color balance [73, 10, 93].

The ability to go beyond detecting faces and automatically label face images with the identity of the persons pictured has a vast number of potential applications. Recent years have seen the development of commercial application of face recognition technology, notably in airport security and online photo-tagging. At the same time, the ubiquity of digital cameras and camcorders and the wealth of images on online social networking sites, combined with the potential for automatic face recognition, has led many to raise potential privacy concerns.

However, both the excitement and fear over widespread application of automatic face recognition may be slightly premature, as there have been notable examples of face recognition systems not performing up to expectations when deployed in commercial applications [68, 23, 26]. To understand why face recognition methods have had mixed success, it is instructive to look at the common databases that were traditionally used to test face recognition algorithms.

One widely used database (that continues to be used) is the Yale B data set [25]. Figure 1.1 shows some representative sample images from Yale B. When comparing these images with a random collection of face images one may encounter in general, such as the images from news photographs in Figure 1.2, a noticeable difference is the uniformity of the images in Yale B. Specifically, all faces are taken from a straight-on frontal pose, with facial features such as eyes in the same position within the image, neutral facial expression, similar lighting condition, and lack of any occluding objects such as hatwear or glasses. This lack of variation from factors such as pose, lighting, expression, and background characterizes many of the standard data sets traditionally used to study face recognition.

Figure 1.1: Sample images from Yale B.

The implicit assumption made by these data sets is the control over the image acquisition process. By controlling image acquisition, one can control aspects such as

Figure 1.2: Sample face images taken from online news photographs. Details of the images are given in Chapter 2.

lighting and background, and instruct the person being photographed to hold a particular pose and expression. This assumption holds for some potential applications of face recognition, such as in security domains where one must prove they are the same person that is pictured in a passport photo. However, for many other applications, this assumption no longer holds, and violating this assumption can lead to rapidly degraded performance.

The central goal of this dissertation is to improve performance on the unconstrained face recognition task, where no control of the image acquisition process is assumed. Doing so first requires establishing a benchmark that accurately reflects unconstrained face recognition performance and that can be used to measure progress. Establishing such a benchmark forms the initial section of this dissertation. The potential value of such a benchmark is that it will provide a well-defined problem that researchers may focus on, as well as a standard metric for assessing performance, which can highlight the current state-of-the-art performance and spur further research. For instance, baseline and initial performance on Caltech 101, a benchmark

for object recognition, was around 15% accuracy in 2004 [33], and current state-of-the-art systems achieve accuracy of more than 75% [24].

Next, we focus on improving unconstrained face recognition by leveraging weakly supervised data that is generally ignored by standard supervised methods, additionally allowing our proposed techniques to easily be applied to recognition and verification tasks on other object classes.

We first put unconstrained face recognition within the broader context of computer vision, and next examine how weakly supervised learning from unlabeled face images can be used to improve face recognition.

1.1 Unconstrained Face Recognition

A fundamental area of research within computer vision is object recognition, which is generally framed as assigning a correct label to an image of an object from a set of known category labels. A canonical data set used in object recognition is Caltech 101 [57], where each image contains one primary object belonging to one of 101 categories, such as ant, beaver, chair, and dollar bill. Object recognition can also be performed at a finer level of granularity, distinguishing between different sub-types of a given class, as in the 102 Category Flowers data set [74], where the category labels are types of flowers such as azalea, buttercup, and carnation.

An important instance of object recognition is face recognition, which has traditionally been studied under an experimental setup referred to as the gallery/probe protocol: at training time, one is presented with n_i images each of N subjects (the gallery), and at test time, given a new probe image, the task is to determine which (if any) of the subjects in the gallery is pictured in the probe image. This protocol was used in databases such as FERET [81] and FRGC [80].

The limitations of this formulation of object recognition are the following two assumptions: there exist only a fixed number of object classes known at training time,

and examples from each class are provided at training time. This is a particularly severe problem with face recognition, since we must re-train the system for every set of identities we wish to be able to recognize, and be provided with training samples of each of these identities.

To remove these assumptions, the task of object recognition can be reformulated as visual verification, where the problem is now to determine, given two images, whether the images are of the same object class (matched pair) or not (mismatched pair). The focus of this dissertation is visual verification applied to unconstrained face images, and we discuss the verification problem formulation in more detail in Chapter 2.

Since the images presented in the test pairs may be of classes not represented in the training set, it is necessary to learn the manner in which an arbitrary object from the set of classes being considered can be transformed from one image to another, due to factors such as viewpoint, background, and occlusions. The large amount of intra-class variability makes the problem of visual identification of never seen objects especially difficult.

As object recognition research has progressed, two issues that have arisen are: how to scale recognition as the number of classes increases; and how to generalize to new categories and quickly learn from a small number of examples. In addition, one of the core difficulties in object recognition is the large amount of intra-class variation in appearance due to factors such as lighting, background, and perspective projection of the 3D object.

Solving the face verification problem requires addressing each of these issues. The verification framework requires generalizing to faces not seen during training, and in face verification the number of identities that a system must be able to distinguish among can become orders of magnitude larger than the typical number of classes used in general object recognition or recognition within a particular category such as

flowers. Addressing the problem of large intra-class variations raises a fundamental issue in computer vision of representation, namely, that an ideal representation should provide discriminative information between classes, yet be invariant or robust to the intra-class variations. This is an especially difficult issue in face recognition, as faces share very similar structure, leading to small inter-class differences, while intra-class variation due to factors such as head pose, background, occlusion, and facial expression can be large.

For these reasons, we believe that progress made on the unconstrained face verification task will also have wider applicability in improving general object recognition. In particular, through weakly supervised learning, as we describe next, the methods presented in this dissertation should have straightforward application to other object categories.

1.2 Weakly Supervised Learning

Generally, face processing is approached using supervised learning. For face recognition, the supervision is in the form of face images labeled with the identity of the person in the image, or pairs of face images that are labeled as two images of the same person or two images of two different persons. In face alignment, the labeled data is often in the form of face images labeled with pose, or the location of facial features such as corners of eyes, nose, and mouth, or training image patches of these specific facial features.

Particularly for face alignment, obtaining this labeled data is manually intensive, and must be repeated for an algorithm to be applied to a new object class outside of faces. In contrast, it is comparatively less effort to obtain many unlabeled face images without identity or pose information. For instance, such images could be obtained by running a face detector over many images, and tuning the detector to produce a low number of false positives (*e.g.*, high precision, low recall). We refer to these unlabeled

face images as partially labeled data, as they have been identified as face images but have no other annotations.

In this dissertation, we focus on making use of the information in this generally unused source of partially labeled data. We make use of unlabeled face images in two ways. First, we show how these images can be automatically jointly aligned with no supervision, and how this can be used to subsequently align additional face images. Second, we show how feature representations can be automatically learned from unlabeled face images, and used in combination with standard image representations to improve verification accuracy.

1.3 Outline

The remainder of the dissertation is organized as follows. In Chapter 2, we present a database for benchmarking performance on the unconstrained face verification task. In Chapter 3, we extend a method for unsupervised joint alignment to work on images of complex objects exhibiting real-world noise. Next, in Chapter 4, we apply unsupervised feature learning using deep learning to improve unconstrained face verification. In Chapter 5, we combine the ideas of unsupervised joint alignment with unsupervised feature learning. We end with conclusions and discussion of potential future work in Chapter 6.

1.4 Contributions

The following are the major contributions made in this dissertation:

1. We present a formulation of the unconstrained face verification problem and create a database for benchmarking performance on this task. This database, Labeled Faces in the Wild, has become widely used in the face recognition community, with over 20 systems evaluated on this data set in published results.

2. We extend the unsupervised joint alignment method of congealing [51], previously only applied to data sets such as hand-written digits, to work on images from complex object classes such as faces and cars. We show that this unsupervised alignment method leads to greater improvement in unconstrained face verification accuracy than a state-of-the-art supervised active appearance model based method.

3. We apply unsupervised feature learning using deep learning to unconstrained face verification. We obtain new image representations that can be combined with representations from hand-crafted image descriptors to achieve state of the art accuracy using a single similarity metric. We develop a local convolutional restricted Boltzmann machine model that is able to take advantage of global structure in an object class while maintaining scalability to high resolution images and robustness to some misalignment.

4. We combine unsupervised joint alignment with unsupervised feature learning, using image representations obtained from deep learning in a congealing framework. We add a sparsity regularization term to the feature learning, causing the learned filters to form a linear topology and improving the quality of the subsequent alignment, as measured in terms of gains in face verification accuracy. Using this unsupervised alignment method, we are able to obtain face verification accuracy matching that obtained through a supervised method based on detecting facial fiducial points.

CHAPTER 2

LABELED FACES IN THE WILD: A DATABASE FOR UNCONSTRAINED FACE VERIFICATION

Most face databases have been created under controlled conditions to facilitate the study of specific parameters on the face recognition problem. These parameters include such variables as position, pose, lighting, background, and camera quality. While there are many applications for face recognition technology in which one can control the parameters of image acquisition, there are also many applications in which the practitioner has little or no control over such parameters. In this chapter, we describe a database, Labeled Faces in the Wild, provided as an aid in studying the latter, unconstrained, recognition problem. The database contains face photographs, labeled with subject names, spanning the range of conditions typically encountered in everyday life. The database exhibits "natural" variability in factors such as pose, lighting, race, accessories, occlusions, and background. In addition to describing the details of the database, we provide specific experimental paradigms for which the database is suitable. This is done in an effort to make research performed with the database as consistent and comparable as possible. We provide baseline results, including results of a state of the art face recognition system combined with a face alignment system. To facilitate experimentation on the database, we provide several parallel databases, including a version in which the faces are more precisely aligned to a common pose, which we shall refer to as the "aligned version".

2.1 Introduction

This chapter describes a database of human face images designed as an aid in studying the problem of *unconstrained face verification*.[1] The database can be viewed and downloaded at `http://vis-www.cs.umass.edu/lfw/`.

Face recognition is the problem of identifying a specific individual, rather than merely detecting the presence of a human face, which is often called *face detection*. The general term "face recognition" can refer to a number of different problems including, but not limited to, the following.

Face Identification: Given a picture of a face, decide which person from among a set of people the picture represents, if any.

Face Verification: Given two pictures, each of which contains a face, decide whether the two people pictured represent the same individual (*e.g.*, verify that the person pictured in one image is the same as the person pictured in the other).

Our database, which we called Labeled Faces in the Wild (LFW), can be used to study these problems in unconstrained environments, as well as other face processing tasks, such as face alignment and face segmentation.

The primary contribution of LFW is providing a large set of relatively unconstrained face images. By unconstrained, we mean faces that show a large range of the variation seen in everyday life. This includes variation in pose, lighting, expression, background, race, ethnicity, age, gender, clothing, hairstyles, camera quality, color saturation, and other parameters. The reason we are interested in natural variation

[1] A note on terminology: For general classes of objects such as cars or dogs, the term "recognition" often refers to the problem of recognizing a member of the larger class, rather than a specific instance. When one "recognizes" a cat (in the context of computer vision research), it is meant that one has identified a particular object as a cat, rather than a particular cat. In the context of recognition of specific instances, as generally referred to when speaking of face recognition, the term *identification* is used to refer to recognizing a specific instance of a class (such as Bob's Toyota) from a set of pre-defined possibilities, as in [21, 41, 22]. The term *verification* is used to refer to verifying that a specific instance of a class in one image is the same specific instance as presented in another image.

is that for many tasks, face recognition must operate in real-world situations where we have little to no control over the composition, or the images are pre-existing. For example, there is a wealth of unconstrained face images on the Internet, and developing recognition algorithms capable of handling such data would be extremely beneficial for information retrieval and data mining. Since LFW closely approximates the distribution of such images, algorithms trained on LFW could be directly applied to web IR applications. In contrast to LFW, existing face databases contain more limited and carefully controlled variation, as we describe in Section 2.2. Figure 2.1 shows images from LFW representative of the diversity in the database. Tables 2.1 gives statistics of LFW such as number of images and people.

LFW is a valuable tool for studying face verification in unconstrained environments, as discussed in Section 2.3. To facilitate fair comparison of algorithms, we give specific protocols for developing and assessing algorithms using LFW (Section 2.4). By construction, algorithm performance on LFW is generalizable to performance in an end-to-end recognition system, as described in Section 2.6. We allow for easy experimentation with LFW by making publicly available parallel versions of the database containing aligned images and superpixel computations (Section 2.7.7). We give baseline results for LFW using both standard and state of the art face recognition methods (Section 2.8).

2.2 Related Databases

There are a number of face databases available to researchers in face recognition. These databases range in size, scope and purpose. The photographs in many of these databases were acquired by small teams of researchers specifically for the purpose of studying face recognition. Acquisition of a face database over a short time and particular location has advantages for certain areas of research, giving the experimenter direct control over the parameters of variability in the database.

Figure 2.1: Sample images from LFW (first row), FRGC (second row), and BioID (third row), representative of variation within each database (best viewed in color).

On the other hand, in order to study more general, unconstrained face recognition problems, in which faces are drawn from a very broad distribution, one should train and test face recognition algorithms on highly diverse sets of faces. While it is possible to manipulate a large number of variables in the laboratory in an attempt to make such a database, there are two drawbacks to this approach. The first is that it is extremely labor intensive. The second is that it is difficult to gauge exactly which distributions of various parameters one should use to make the most useful database. What percentage of subjects should wear sunglasses, or have beards, or be smiling? How many backgrounds should contain cars, boats, grass, deserts, or basketball courts?

One possible solution to this problem is simply to measure a "natural" distribution of faces. Of course, no single canonical distribution of faces can capture a natural distribution that is valid across all possible application domains. Our database uses a set of images that was originally gathered from news articles on the web. This set clearly has its own biases. For example, there are not many images which occur under very poor lighting conditions. Also, because we use the Viola-Jones detector as a filter for the database, there are a limited number of side views of faces, and few views from

Database	# of people	Total images
LFW	5749	13233
FRGC	>466	>50000
BioID	23	1521
FERET	1199	14126

(a) Comparison of LFW, FRGC, and BioID

# of images /person	# of people (% of people)	# of images (% of images)
1	4069 (70.8)	4096 (30.7)
2-5	1369 (23.8)	3739 (28.3)
6-10	168 (2.92)	1251 (9.45)
11-20	86 (1.50)	1251 (9.45)
21-30	25 (0.43)	613 (4.63)
31-80	27 (0.47)	1170 (8.84)
> 81	5 (0.09)	1140 (8.61)
Total	5749	13233

(b) Distribution of LFW

Table 2.1: Face Database Statistics

above or below. However, the range and diversity of pictures present is very large. We believe such a database will be an important tool in studying unconstrained face recognition.

Existing face databases generally differ from LFW in one of two key aspects. Labeled databases for recognition, such as the Face Recognition Grand Challenge [80], BioID [44], FERET [70], and CMU PIE [100], are typically taken under very controlled conditions, with fewer people and less diversity than LFW. For instance, images in LFW often contain complex phenomena such as headgear, additional people or faces in the background, and self-occlusion. Moreover, variations in parameters such as pose, lighting, and expression are carefully controlled in other databases, as compared with the uncontrolled variation in LFW that approximates the conditions in every day life. On the other hand, databases such as Caltech 10000 Web Faces [1] present

highly diverse image sets similar to LFW, but are designed for face detection and do not contain person labels, making them unsuitable for recognition.

We now discuss the origin for LFW and comparisons with two of the more similar existing face recognition databases.[2]

2.2.1 Faces in the Wild

The Faces in the Wild project [6],[5] demonstrated that a large, partially labeled database of face images could be built using imperfect data from the web.[3] The database was built by jointly analyzing pictures and their associated captions to cluster images by identity. The resulting data set, which achieved a labeling accuracy of 77% [5], was informally referred to as "Faces in the Wild".

However, the database was not intended to act as training and test data for new experiments, and contained a high percentage of label errors and duplicated images. As a result, various researchers derived ad hoc subsets of the database for new research projects [41, 35, 78, 75]. The need for a clean version of the data set warranted doing the job thoroughly and publishing a new database, which resulted in Labeled Faces in the Wild.

2.2.2 Face Recognition Grand Challenge Databases

The Face Recognition Grand Challenge (FGRC) [80] was designed to study the effect of new, richer data types on face recognition, and thus includes high resolution data, three-dimensional scans, and image sequences. In contrast, LFW consists of faces extracted from previously existing images and hence can be used to study recognition from images that were not taken for the special purpose of face recognition by machine.

---

[2]See [38] for more detailed comparisons and a more complete list of existing face databases.

[3]Note this is not the same as *Labeled* Faces in the Wild.

Another important difference between the data sets associated with the FRGC and our data set is the general variety of images. For example, while there are large numbers of images with uncontrolled lighting in the FRGC data sets, these images contain a great deal less natural variation than the LFW images. For example, the FRGC outdoor uncontrolled lighting images contain two images of each subject, one smiling and one with a neutral expression. The LFW images, in contrast contain arbitrary expressions. Variation in clothing, pose, background, and other variables is much greater in LFW than in the FRGC databases. As mentioned earlier, the difference is one of *controlled variation* (FRGC) versus *natural* or *random* variation (LFW).

2.2.3 BioID Face Database

Similar to LFW, the BioID Face Database [44] strives to capture realistic settings with variability in pose, lighting, and expression. Unlike LFW, however, the distribution of images is more limited, focusing on a small number of home and office environments. Images for a given individual are generally different views of the same scene, whereas images in LFW for a given individual tend to be from a variety of venues. In addition, LFW has much more variability with respect to race, as the large majority of people in BioID are Caucasians. Finally, BioID is targeted at the face detection problem, and no person labels are given, so images would need to be manually labeled to be used for recognition.

While BioID is an interesting database of face images which may be useful for a number of purposes such as face detection in indoor environments, LFW will be useful for solving more general and difficult face recognition problems with large populations in highly variable environments.

In summary, there are a great number of face databases available, and while each has a role in the problems of face recognition or face detection, LFW fills an important gap for the problem of unconstrained face recognition.

2.3 Intended Uses

As mentioned in the introduction, this database is aimed at studying face recognition in realistic, unconstrained environments. Specifically, we focus on the unconstrained face verification problem, in contrast to the traditional gallery/probe face identification set-up. In this set-up, there is a pre-specified gallery consisting of face images of a set of people, where the identity of each face image is known. The problem is to take a new query image, and decide which person in the gallery the new image represents. For instance, the gallery may consist of 10 images each of 10 different people, and the task would be to decide which of the 10 people a new input image represents.

Generally, face verification has been tested in situations where both the gallery images and query images are taken under controlled environments. For instance, even in Experiment 4 of the FRGC [80], which was designed to test the case in which the query images are taken in a more uncontrolled environment, the gallery images are still controlled.

The assumption of pre-defined gallery images is reasonable for certain tasks, such as recognition for security access, where the images can be taken ahead of time in a fixed environment, and query images can be taken in the same environment. On the other hand, for a large range of tasks, this assumption does not hold. For instance, as an information retrieval task, a user may wish to have photos automatically tagged with the names of the people, using a gallery of previously manually annotated photographs, which would not be taken in a controlled environment. Therefore, we focus on using LFW to study the following unconstrained face verification problem.

2.3.1 Unconstrained Face Verification

An alternative formulation of face recognition to the gallery/probe set-up is the pair matching face verification paradigm: given a pair of face images, decide whether the images are of the same person. Within the pair matching paradigm, there are a number of subtly, but importantly different recognition problems. Some of these differences concern the specific organization of training and testing subsets of the database. A critical aspect of our database is that for any given training-testing split, the people in each subset are mutually exclusive. In other words, for any pair of images in the training set, neither of the people pictured in those images is in any of the test set pairs. Similarly, no test image appears in a corresponding training set. We refer to this case, in which neither of the individuals pictured in the test pair have been seen during training, as the *unseen pair match* problem.

At training time, it is essentially impossible to build a model for any person in the test set, making this problem substantially different from the gallery/probe paradigm. In particular, for LFW, since the people in test images have never been seen before, there is no opportunity to build models for such individuals, except to do this at test time from a single image. Instead, this paradigm is meant to focus on the generic problem of differentiating any two individuals that have never been seen before. Thus, a different type of learning is suggested: learning to discriminate among any pair of faces, rather than learning to find exemplars of a gallery of people as in face verification. Recently, there have been several important developments in this area of face recognition research [21, 75, 41].

A closely related problem to unseen pair matching is learning from one example [8], although there are subtle differences between the two:

- In learning from one example (per person), training examples are given at training time. Whereas in the unseen pair match problem, the single model image is not available until test time. If processing speed is an important constraint,

then it may be advantageous to have a training example ahead of time, as in the learning from one example paradigm.

- Another important difference is that in learning from one example, at test time, the objective is usually to determine which, if any, of the models the test image corresponds to. One would not normally identify the test image with more than one model, and so a winner-take-all or maximum likelihood approach for selecting a match would be reasonable. On the other hand, in the unseen pair match problem, the objective is to make a binary decision about whether a given single image matches another image. If a test set contains multiple pairings of a single image B, i.e., a group of pairs of images of the form $(A_i, B), 1 \leq i \leq n$, there is no mechanism for deciding that the image B should match only one of the images A_i. In other words, each pairwise decision is made independently. This rules out the winner-take-all or maximum likelihood style approaches.

2.3.2 Face Identification versus Face Verification

As mentioned earlier, we believe that face verification under the unseen pair matching formulation is one of the most general and fundamental face recognition problems. At a basic level, human beings are capable of recognizing faces after only seeing one example image, and thus are fundamentally different from algorithms that are only capable of performing matching against a fixed gallery of exemplars. Moreover, as recognition systems are scaled to attempt to deal with orders of magnitude more people, algorithms designed to learn general variability will be less computationally and resource intensive than methods that attempt to learn a specific model for each person.

From a practical standpoint, pair matching algorithms require less supervision, only requiring examples of matching and mismatching pairs, rather than exemplars of each person to be identified. For instance, this would significantly simplify the

previously mentioned image annotation problem. A pair matching algorithm could be trained independently on separate existing data, then used to label photographs in a collection with the names of the people pictured by clustering face images that were likely to be the same person. In comparison, a face verification algorithm would require manually labeled examples and would only be able to recognize from among the people appearing in the labeled examples.

For these reasons, we believe the unseen pair matching problem is an important area of face recognition and that having the LFW database as a benchmark for developing and comparing algorithms will help push new developments in this area. In addition to containing a larger variety of images matching real-life complexity than existing databases, LFW also contains a larger number of people, an important aspect for pair matching, allowing algorithms to discriminate between general faces rather than a specific small number of faces within a gallery.

2.4 Protocols

Proper use of training, validation, and testing sets is crucial for the accurate comparison of face recognition algorithms. For instance, performance will be improperly biased upward if the parameters of the algorithm are inadvertently tuned to the test set. We provide clear guidelines for the use of this data to minimize "fitting to the test data". Also, the size and difficulty of the data set may mitigate the degree to which unintended overfitting problems may occur.

We organize our data into two "Views", or groups of indices. View 1 is for algorithm development and general experimentation, prior to formal evaluation. This might also be called a model selection or validation view. View 2, for performance reporting, should be used only for the final evaluation of a method. The goal of this methodology is to use the final test sets as seldom as possible before reporting.

2.4.1 View 1: Model selection and algorithm development

The main purpose of this view of the data is so that researchers can freely experiment with algorithms and parameter settings without worrying about overusing test data. For example, if one is using support vector machines and trying to decide upon which kernel to use, it would be appropriate to test various kernels on View 1 of the database. Training and testing algorithms from this view may be repeated as often as desired without significantly biasing final results.

2.4.2 View 2: Performance reporting

The second view of the data should be used sparingly, and only for performance reporting. Ideally, it should only be used once, as choosing the best performer from multiple algorithms, or multiple parameter settings, will bias results toward artificially high accuracy. Once a model or algorithm has been selected (using View 1 if desired), the performance of that algorithm can be measured using View 2. For both recognition paradigms, View 2 consists of 10 splits of training and test sets, and the experimenter should report aggregate performance of a classifier on these 10 separate experiments.

It is critical for performance reporting that the final parameters of the classifier under each experiment be set using either the data in View 1 or only the training data for that experiment. An algorithm may not, during performance reporting, set its parameters to maximize the combined accuracy across all 10 training sets. The training and testing sets overlap across experiments, thus optimizing a classifier simultaneously using all training sets is essentially fitting to the test data, since the training set for one experiment is the testing data for another. In other words, each of the 10 experiments (both the training and testing phases) should be run completely independently of the others, resulting in 10 separate classifiers (one for each test set).

While there are many methods for reporting the final performance of a classifier, including receiver operating characteristic (ROC) curves and Precision-Recall curves, we ask that each experimenter, at a minimum, report the **estimated mean accuracy** and the **standard error of the mean** for View 2 of the database. The estimated mean accuracy is $\hat{\mu} = \sum_{i=1}^{10} p_i/10$, where p_i is the percentage of correct classifications on subset i of View 2. It is important to note that accuracy should be computed with parameters and thresholds chosen independently of the test data, ruling out, for instance, simply choosing the point on a precision-recall curve giving the highest accuracy. The standard error of the mean is $S_E = \hat{\sigma}/\sqrt{10}$, where $\hat{\sigma}$ is the estimate of the standard deviation, $\hat{\sigma} = \sqrt{\sum_{i=1}^{10}(p_i - \hat{\mu})^2/9}$.

The training sets in View 2 overlap, therefore the standard error may be biased downward somewhat relative to what would be obtained with fully independent training sets and test sets. However, because the test sets of View 2 are independent, we believe this quantity will be valuable in assessing the significance of the difference among algorithms.[4]

View 1 of LFW consists of two subsets of the database, one for training, containing 2200 pairs, and one for testing, containing 1000 pairs. The persons appearing in the training and testing sets are mutually exclusive. View 2 consists of 6000 pairs, divided into ten subsets, and performance is computed using 10-fold cross validation using those subsets.

It should be noted that some images in View 1 may appear in View 2 as well, as the two views were selected randomly and independently from the entire database. This multiple-view approach has been used, rather than a traditional training-validation-testing split of the database, in order to maximize the amount of data available for

[4]To determine if the difference in performance between two algorithms is statistically significant at the 0.05 level, one should compute confidence intervals of 85% for the mean accuracy of each algorithm and test if these intervals overlap [79].

training and testing. Ideally, one would have enough images in a database so that training, validation, and testing sets could be non-overlapping. However, in order to maximize the size of our training and testing sets, we have allowed reuse of the data between View 1 of the database and View 2 of the database. The bias introduced into the results by this approach is very small and outweighed by the benefit of the resulting larger training and test set sizes. (Unfortunately, this data reuse between View 1 and View 2 has the potential for inadvertent overfitting by inappropriate use of View 1. We mention this issue again in Chapter 4, and discuss it further in Appendix A.)

2.5 Transitivity and the Image-Restricted and Unrestricted Use of Training Data

Whenever one works with matched and mismatched data pairs, the issue of creating auxiliary training examples by using the transitivity of equality arises. For example, in a training set, if one matched pair consists of the 10th and 12th images of George_W_Bush, and another pair consists of the 42nd and 50th images of George_W_Bush, then it might seem reasonable to add other image pairs, such as (10, 42), (10, 50), (12, 42) and (12, 50), to the training data using an automatic procedure. One could argue that such pairs are *implicitly present* in the original training data, given that the images have been labeled with the name George_W_Bush. Auxiliary examples could be added to the mismatched pairs using a similar method.

Rather than disallowing such augmentation or penalizing researchers who do not wish to add many thousands of extra pairs of images to their training sets, we give two separate methods for using training data. When reporting results, the experimenter should state explicitly whether the *image-restricted* or the *unrestricted* training method was used to generate results. These two methods of training are described next.

2.5.1 Image-Restricted Training

The idea behind the image-restricted paradigm is that the experimenter should *not* use the name of a person to infer the equivalence or non-equivalence of two face images that are not explicitly given in the training set. Under the image-restricted training paradigm, the experimenter should discard the actual names associated with a pair of training images, and retain only the information about whether a pair of images is matched or mismatched. Thus, if the pairs (10,12) and (42,50) of George_W_Bush are both given explicitly in a training set, then under the image-restricted training paradigm, there would be no simple way of inferring that the 10th and 42nd images of George_W_Bush were the same person, and thus this image pair should not be added to the training set.

Note that under this paradigm, it is still possible to augment the training data set by comparing *image similarity*, as opposed to name equivalence. For example, if the 1st and 2nd images of a person form one matched training pair, while the 2nd and 3rd images of the same person form another matched training pair, one could infer from the *equivalence of images* in the two pairs that the 1st and 3rd images came from the same person, and add this pair to the training set as a matched pair. Such image-based augmentation is allowed under the image-restricted training paradigm.

Both Views of the database support the image-restricted training paradigm. In View 1 of the database, the file `pairsDevTrain.txt` is intended to support the image-restricted use of training data, and `pairsDevTest.txt` contains test pairs. In View 2 of the database, the file `pairs.txt` supports the image-restricted use of training data. Formats of all such files are given in Section 2.7.6.

2.5.2 Unrestricted Training

The idea behind the unrestricted training paradigm is that one may form as many pairs of matched and mismatched pairs as desired from a set of images labeled with

individuals' names. To support this use of the database, we defined subsets of *people*, rather than image pairs, that can be used as a basis for forming arbitrary pairs of matched and mismatched images.

In View 1 of the database, the files `peopleDevTrain.txt` and `peopleDevTest.txt` can be used to create arbitrary pairs of training and testing images. For example, to create mismatched training pairs, choose any two people from `peopleDevTrain.txt`, choose one image of each person, and add the pair to the data set. Pairs should *not* be constructed using mixtures of images from training and testing sets.

In View 2 of the database, the file `people.txt` supports the unrestricted training paradigm. Training pairs should be formed only using pairs of images from the same subsets. Thus, to form a training pair of mismatched images, choose two people from the same subset of people, choose an image of each person, and add the pair to the training set. Note that in View 2 of the database, which is intended only for performance reporting, the test data is fully specified by the file `pairs.txt`, and should not be constructed using the unrestricted paradigm. The unrestricted paradigm is only for use in creating *training* data.

Due to the added complexity of using the unrestricted paradigm, we suggest that users start with the image-restricted paradigm by using the pairs described in `pairsDevTrain.txt`, `pairsDevTest.txt`, and, for performance reporting, `pairs.txt`. Later, if the experimenters believe that that their algorithm may benefit significantly from larger amounts of training data, they may wish to consider using the unrestricted paradigm. In either case, it should be made clear in any publications which training paradigm was used to train classifiers for a given test result.

2.6 The Detection-Alignment-Recognition Pipeline

Many real world applications wish to automatically detect, align, *and* recognize faces in a larger still image, or in a video of a larger scene. Thus, face recognition

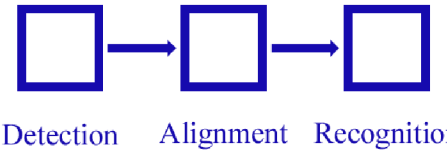


Figure 2.2: The Detection-Alignment-Recognition (DAR) pipeline. The images of the LFW database represent the output of the Viola-Jones detector. By working with such a database, the developer of alignment and recognition algorithms know that their methods will fit easily into the DAR pipeline.

is often naturally described as part of a Detection-Alignment-Recognition (DAR) pipeline, as illustrated in Figure 2.2. To complete this pipeline, we need automatic algorithms for each stage of the pipeline. In addition, each stage of the pipeline must either accept images from, or prepare images for, the next stage of the pipeline. To facilitate this process, we have purposefully designed our database to represent the output of the detection process.

In particular, every face image in our database is the output of the Viola-Jones face detection algorithm [108]. The motivation for this is as follows. If one can develop a face alignment algorithm (and subsequent recognition algorithm) that works directly on LFW, then it is likely to also work well in an end-to-end system that uses a face detector as a first step. This alleviates the need for each researcher to worry about the process of detection, or about the possibility that a manually aligned database does not adequately represent the true variability seen in the world. In other words, it allows the experimenter to focus on the problems of alignment and recognition rather detection.

2.7 Construction and Composition Details

The process of building the database can be broken into the following steps:

1. gathering raw images,

2. applying a face detector and manually eliminating false positives,

3. eliminating duplicate images,

4. hand labeling (naming) the detected people,

5. cropping and rescaling the detected faces, and

6. forming pairs of training and testing pairs for View 1 and View 2 of the database.

2.7.1 Gathering raw images

As a starting point, we used the raw images from the Faces in the Wild database collected by Tamara Berg at Berkeley. Details of this set of images can be found in [6].

2.7.2 Detecting faces

A version of the Viola-Jones face detector [108] was run on each image. Specifically, we used the code in OpenCV, version 1.0.0, release 1. Faces were detected using the function `cvHaarDetectObjects`, using the provided Haar classifier cascade `haarcascade_frontalface_default.xml`, with scale_factor set to 1.2, min_neighbors set to 2, and the flag set to `CV_HAAR_DO_CANNY_PRUNING`.

For each positive detection (if any), the following procedure was performed:

1. If the highlighted region was determined by the operator to be a non-face, it was omitted from the database.

2. If the name of the person of a detected face from the previous step could not be identified, either from general knowledge or by inferring the name from the associated caption, then the face was omitted from the database.

3. If the same picture of the same face was already included in the database, the face was omitted from the database. More details are given below about eliminating duplicates from the database.

4. Finally, if all of these criteria were met, the face was recropped and rescaled (as described below) and saved as a separate JPEG file.

2.7.3 Eliminating duplicate face photos

A good deal of effort was expended in removing duplicates from the database. While we considered including duplicates, since it could be argued that humans may often encounter the exact same picture of a face in advertisements or in other venues, ultimately it was decided that they would prove to be a nuisance during training in which they might cause overfitting of certain algorithms. In addition, any researcher who chooses may easily add duplicates for his or her own purposes, but removing them is somewhat more tedious.

Before removing duplicates, it is necessary to define exactly what they are. While the simplest definition, that two pictures are duplicates if and only if the images are numerically equivalent at each pixel, is somewhat appealing, it fails to capture large numbers of images that are indistinguishable to the human eye. We found that the unfiltered database contained large numbers of images that had been subtly recropped, rescaled, renormalized, or variably compressed, producing pairs of images which were visually nearly equivalent, but differed significantly numerically.

We chose to define duplicates as images which were judged to have a common original source photograph, irrespective of the processing they had undergone. While we attempted to remove all duplicates as defined above from the database, there may exist some remaining duplicates that were not found. We believe the number of these is small enough so that they will not significantly impact research. The problem of near-duplicate detection has also been studied by Jain and Learned-Miller [42], where a semi-automatic method was developed to identify near-duplicates.

In addition, there remain a number of pairs of pictures which are extremely similar, but clearly distinct. For example, there appeared to be pictures of celebrities

taken nearly simultaneously by different photographers from only slightly different angles. Whenever there was evidence that a photograph was distinct from another, and not merely a processed version of another, it was maintained as an example in the database.

2.7.4 Labeling the faces

Each person in the database was named using a manual procedure that used the caption associated with a photograph as an aid in naming the person. It is possible that certain people have been given incorrect names, especially if the original news caption was incorrect. Following the release of the database, a small number of labeling errors have been discovered (see Section 2.10).

Significant efforts were made to combine all photographs of a single person into the same group under a single name. This was at times challenging, since some people showed up in the original captions under multiple names, such as "Bob McNamara" and "Robert McNamara". When there were multiple possibilities for a person's name, we strove to use the most commonly seen name for that person. For Chinese and some other Asian names, we maintained the common Chinese ordering (family name followed by given name), as in "Hu Jintao". Note that there are some people in the database with just a single name, such as "Abdullah" or "Madonna". There is also one case of two people with the same name, "Jim O'Brien"; however, these two people were mistakenly labeled as being the same person.

2.7.5 Cropping and rescaling

For each labeled face, the final image to place in the database was created using the following procedure. The region returned by the face detector for the given face was expanded by 2.2 in each dimension. If this expanded region would fall outside the original image area, then a new image of size equal to the desired expanded region was created, containing the corresponding portion of the original image but padded with

black pixels to fill in the area outside the original image. The expanded region was then resized to 250 by 250 pixels using the function `cvResize`, in conjunction with `cvSetImageROI` as necessary. The images were then saved in the JPEG 2.0 format.

2.7.6 Forming training and testing sets

Forming sets and pairs for View 1 and View 2 was done using the following process. First, each specific person in the database was randomly assigned to a set. In the case of View 1, each person had a 0.7 probability of being placed into the training set, and in the case of View 2, each person had a uniform probability of being placed into each set.

The people in each set are given in `peopleDevTrain.txt` and `peopleDevTest.txt` for View 1 and `people.txt` for View 2. The first line of `peopleDevTrain.txt` and `peopleDevTest.txt` gives the total number of people in the set, and each subsequent line contains the name of a person followed by the number of images of that person in the database. `people.txt` is formatted similarly, except that the first line gives the number of sets. The next line gives the number of people in the first set, followed by the names and number of images of people in the first set, then the number of people in the second set, and so on for all ten sets.

Matched pairs were formed as follows. First, from the set of *people* with at least two images, a person was chosen uniformly at random (people with more images were given the same probability of being chosen as people with fewer images). Next, two images were drawn uniformly at random from among the images of the given person. If the two images were identical or if the pair of images of the specific person was already chosen previously as a matched pair, then the whole process was repeated. Otherwise the pair was added to the set of matched pairs.

Mismatched pairs were formed as follows. First, from the set of *people* in the set, two people were chosen uniformly at random (if the same person was chosen twice

then the process was repeated). One image was then chosen uniformly at random from the set of images for each person. If this particular image pair was already chosen previously as a mismatched pair, then the whole process was repeated. Otherwise the pair was added to the set of mismatched pairs.

The pairs for each set are given in `pairsDevTrain.txt` and `pairsDevTest.txt` for View 1 and `pairs.txt` for View 2. The first line of `pairsDevTrain.txt` and `pairsDevTest.txt` gives the total number N of matched pairs (equal to the total number of mismatched pairs) in the set. The next N lines give the matched pairs in the format.

```
name    n1    n2
```

which means the matched pair consists of the `n1` and `n2` images for the person with the given name. For instance,

```
George_W_Bush    10    24
```

would mean that the pair consists of images `George_W_Bush_0010.jpg` and `George_W_Bush_0024.jpg`.

The following N lines give the mismatched pairs in the format

```
name1    n1    name2    n2
```

which means the mismatched pair consists of the `n1` image of person `name1` and the `n2` image of person `name2`. For instance,

```
George_W_Bush    12    John_Kerry    8
```

would mean that the pair consists of images `George_W_Bush_0012.jpg` and `John_Kery_0008.jpg`.

The file `pairs.txt` is formatted similarly, except that the first line gives the number of sets followed by the number of matched pairs N (equal to the number of

Figure 2.3: Examples of superpixels. The left column is the original image, the middle column is the Mori segmentation (`N_sp`=100, `N_sp2`=200, `N_ev`=40), and the right column is the Felzenszwalb-Huttenlocher segmentation (`sigma`=0.5, `K`=100, `min`=20).

mismatched pairs). The next $2N$ lines give the matched pairs and mismatched pairs for set 1 in the same format as above. This is then repeated nine more times to give the pairs for the other nine sets.

2.7.7 Parallel Databases

To facilitate experimentation on LFW, we also present several parallel versions of our database. We created an aligned version of the database, and for both the original and the aligned versions, we computed superpixels for each image.

To create an aligned version of our database, we used an implementation of the congealing and funneling method described next in Chapter 3 [35].[5] We took one image each of 800 people selected at random from View 1 to learn a sequence of distribution fields, which we then used to funnel every image in the database.

A superpixel representation of an image is a division of the image into a number of small contiguous regions where the pixel values in each region are homogeneous. It is thus a type of oversegmentation of an image. Superpixels have recently started replacing pixels as the basic building block for an image in several object recognition

[5] `http://vis-www.cs.umass.edu/code/congealingcomplex/`

method	database	$\hat{\mu} \pm S_E$
Eigenfaces	unaligned	0.6002 ± 0.0079
Nowak	unaligned	0.7245 ± 0.0040
Nowak	funneled	0.7333 ± 0.0060

Table 2.2: Accuracy on View 2

and segmentation models [67, 32, 92, 2].[6] This transition is partly due to the larger spatial support that superpixels provide, allowing more global features to be computed than on pixels alone.

Superpixel representations have already been successfully applied to face segmentation [2] and we believe they can also be useful for detection and recognition. Therefore, we provide superpixel representations for all the images in the database based on Mori's online implementation [67].[7] We also experimented with the Felzenszwalb and Huttenlocher [20][8] algorithm but found that Mori's method, while more computationally expensive, did a much better job at preserving the face-background boundary, a crucial property for superpixel-based segmentation. Figure 2.3 contains sample superpixel results of both methods on four diverse images from the database.

2.8 Results

To establish baseline results as well as validate the difficulty of LFW, we used the standard face recognition method of Eigenfaces [106]. We computed eigenvectors from the training set of View 1 and determined the threshold value for classifying pairs as matched or mismatched that gave the best performance on the test set of View 1.

[6]While the term superpixels has only recently been defined, the idea of using oversegmentations has existed in the vision community dating back to at least 1989 [7].

[7]`http://www.cs.sfu.ca/~mori/research/superpixels/`

[8]`http://people.cs.uchicago.edu/~pff/segment/`

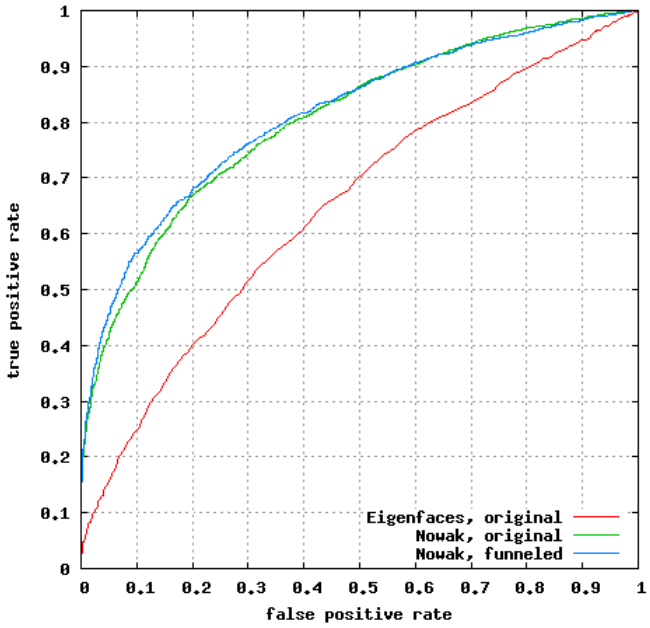


Figure 2.4: ROC curves for pair matching

For each run of View 2, the training set was used to compute the eigenvectors, and pairs were classified using the threshold on Euclidean distance from View 1.

To determine the best performance on pair matching, we ran an implementation[9] of the recognition system of Nowak and Jurie [75], which was state of the art at the time of the release of LFW. The Nowak algorithm gives a similarity score to each pair, and View 1 was used to determine the threshold value for classifying pairs as matched or mismatched. For each of the 10 folds of View 2 of the database, we trained on 9 of the sets and computed similarity measures for the held out test set, and classified pairs using the threshold.

We also ran the Nowak algorithm on the parallel aligned database of LFW, again using View 1 to pick the threshold that optimized performance on the test set.

[9] `http://lear.inrialpes.fr/people/nowak/similarity/index.html`

The mean classification accuracy $\hat{\mu}$ and the standard error of the mean S_E are given in Table 2.2. In addition, the mean ROC curves for pair matching are given in Figure 2.4. Each point on the curve represents the average over the 10 folds of (false positive rate, true positive rate) for a fixed threshold.

Chance performance is 0.5 on the pair matching task. The low accuracy of Eigenfaces reflects the difficulty of the images in LFW and of unconstrained face recognition in general. While the Nowak method significantly outperforms Eigenfaces, it is still far below estimated human-level performance (see Section 2.10) and there is a large amount of room for improvement.

Comparing the accuracy between the Nowak recognizer on the unaligned and funneled images, the standard errors of the mean overlap. Therefore, the difference between the two is not statistically significant. Nonetheless, combining the Nowak recognition system out of the box with the funneling alignment provides a higher baseline to compare against. In addition, judging from the ROC curves, the advantage of using the aligned images may be more pronounced for a cost function emphasizing higher true positive rate at lower false positive rates of approximately 0.1. As a general comment, while simply running an algorithm on the aligned database is likely to improve performance over the same algorithm on the original database, modifying the algorithm to take advantage of the tighter correspondence of faces in the aligned version can potentially do even better.

2.9 Discussion

We have created a set of resources for researchers interested in unconstrained face recognition. Specifically, we have

1. Introduced a new labeled database, Labeled Faces in the Wild, that contains 13,233 images of 5749 unique individuals with highly variable image conditions. The natural variability and difficulty of this database allows models learned

to be applied to new unseen images (taken from the web, for example). This database also fits neatly into the Detection-Alignment-Recognition pipeline.

2. Devised model selection and performance reporting splits for the face verification task. The splits and suggested evaluation metrics were designed to facilitate fair comparisons of algorithms and avoid inadvertently overfitting to the test data.

3. Provided baseline results using Eigenfaces, both as an example of how to set algorithm parameters and to validate the difficulty of this database for both recognition problems.

4. Provided results using the state of the art (at the time of the release of LFW) method [75] for pair matching.

5. Provided parallel versions of the database. The aligned version can be used to improve the performance and run time (by reducing the search space) and computed superpixels preserve the face-background boundary well and can be reliably used for detection, recognition, and segmentation.

2.10 History of LFW After Release

After the release of LFW, a small number of labeling errors were discovered.[10] The decision was made to freeze the database in its original form, and require methods evaluated on LFW to use the labels as originally given, so that all published results would be consistent.

Kumar *et al.* [49] estimated human performance on LFW, using Amazon Mechanical Turk and asking each person to rate their confidence that the pair of images presented belonged to the same person or not. Using the full LFW images, human

[10] `http://vis-www.cs.umass.edu/lfw/#errata`

performance was estimated at 0.9920, indicating a significant gap between human and machine accuracy.[11] Additionally, they performed the same experiment where the face region, encompassing the eyes, nose, and mouth, were masked out, and human performance only dropped to 0.9427. The authors suggested that this implies performance on LFW may be inflated by making use of information contained in the background (*e.g.*, if multiple images of the same person were taken at the same event). However, as the mask leaves certain regions of the person visible, such as hair and chin, an alternate interpretation is that there remains useful information for discriminating between people in these regions that are often ignored by machine verification systems.

To date, LFW has been cited in over 200 publications,[12] and 20 methods have been evaluated on LFW, of which three have presented results under the unrestricted training protocol. For the methods using the image-restricted training protocol, we further divided the methods into categories based on the amount of training data used that was outside of LFW. We roughly grouped these into methods that made no use of training data outside of LFW, methods that made implicit use of outside training data in the form of trained facial feature detectors (used either to align the images as in LFW-a or to determine where to extract features from in an image), and finally methods that made explicit use of outside training data in the recognition system itself.

Tables 2.3, 2.4, and 2.5 give the accuracy for method using the image-restricted training protocol, for each of these three divisions. Table 2.6 gives the accuracy for

---

[11] As LFW was created from news photographs, human performance on LFW will also reflect a person's previous knowledge of the persons shown in the LFW images, *e.g.*, famous celebrities. To obtain an estimate on human performance on LFW, limited to the unfamiliar faces a person has not seen before, assume that the set of already familiar faces accounts for a fraction α of LFW. Performance on the unfamiliar faces can then be estimated as $\frac{0.9920-\alpha}{1-\alpha}$. For a conservative estimate based on a large α of 0.5, performance on unfamiliar faces is 0.9840, still significantly higher than machine performance.

[12] As indicated by Google Scholar: `http://scholar.google.com/`

	$\hat{\mu} \pm S_E$
Eigenfaces, [106], 1991	0.6002 ± 0.0079
Nowak, [75], 2007	0.7245 ± 0.0040
Nowak on funneled images, [35], 2007	0.7393 ± 0.0049
Hybrid descriptor-based, [111], 2008	0.7847 ± 0.0051
Multi-Region Histograms, [97], 2009	0.7295 ± 0.0055
Pixels/MKL, [85], 2009	0.6822 ± 0.0041
V1-like/MKL, [85], 2009	0.7935 ± 0.0055

Table 2.3: LFW verification accuracy for methods trained using the image-restricted protocol, with no use of training data outside LFW.

	$\hat{\mu} \pm S_E$
MERL, [36], 2008	0.7052 ± 0.0060
MERL+Nowak, [36], 2008	0.7618 ± 0.0058
LDML, [29], 2009	0.7927 ± 0.0060
Hybrid, [104], 2009	0.8398 ± 0.0035
Combined b/g samples based methods, [112], 2009	0.8683 ± 0.0034
Single LE + holistic, [11], 2010	0.8122 ± 0.0053
LARK supervised, [99], 2011	0.8510 ± 0.0059
DML-eig SIFT, [115], 2012	0.8127 ± 0.0230
DML-eig combined, [115], 2012	0.8565 ± 0.0056

Table 2.4: LFW verification accuracy for methods trained using the image-restricted protocol, using training data outside LFW for alignment or feature extraction.

	$\hat{\mu} \pm S_E$
Attribute classifiers, [49], 2009	0.8362 ± 0.0158
Simile classifiers, [49], 2009	0.8414 ± 0.0131
Attribute and Simile classifiers, [49], 2009	0.8529 ± 0.0123
NReLU, [69], 2010	0.8073 ± 0.0134
Multiple LE + comp, [11], 2010	0.8445 ± 0.0046
Associate-Predict, [114], 2011	0.9057 ± 0.0056

Table 2.5: LFW verification accuracy for methods trained using the image-restricted protocol, using training data outside LFW in recognition system (beyond alignment and feature extraction).

	$\hat{\mu} \pm S_E$
LDML-MkNN, [29], 2009	0.8750 ± 0.0040
Combined multishot, [104], 2009	0.8950 ± 0.0051
LBP multishot, [104], 2009	0.8517 ± 0.0061
LBP PLDA, [58], 2012	0.8733 ± 0.0055
combined PLDA, [58], 2012,	0.9007 ± 0.0051

Table 2.6: LFW verification accuracy for methods trained using unrestricted protocol.

methods using the unrestricted protocol. Figures 2.5, 2.6, and 2.7 show the ROC curves on LFW for these methods.

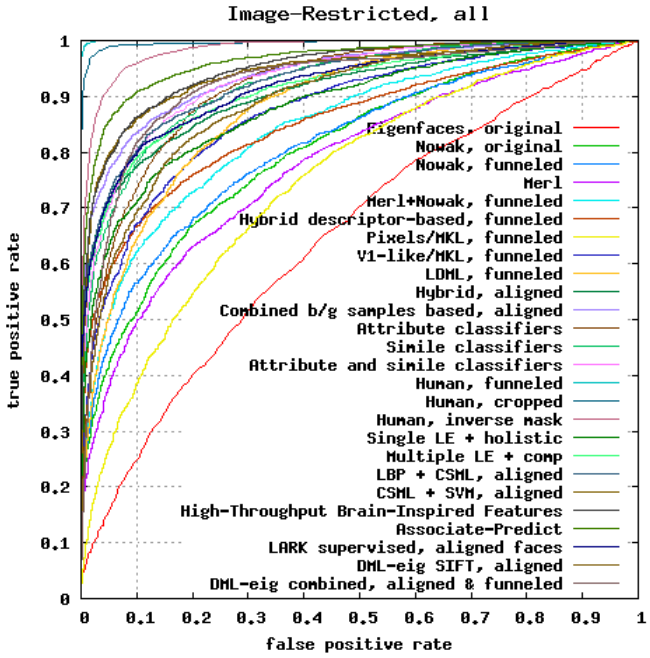


Figure 2.5: ROC curves on LFW for methods trained using the image-restricted protocol.

Underscoring the difficulty of unconstrained face verification, the baseline of Eigenfaces, which gives reasonable performance on data sets such as Yale, gives significantly worse performance than the more recent methods evaluated on LFW. Additionally, the method of V1-like features, which gives 80% accuracy on LFW, yields over 98%

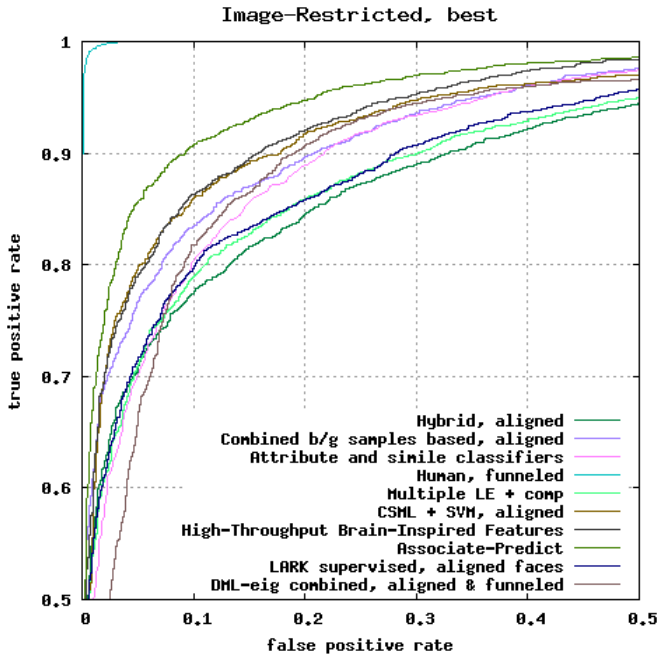


Figure 2.6: ROC curves on LFW for a subset of the highest-accuracy methods trained using the image-restricted protocol.

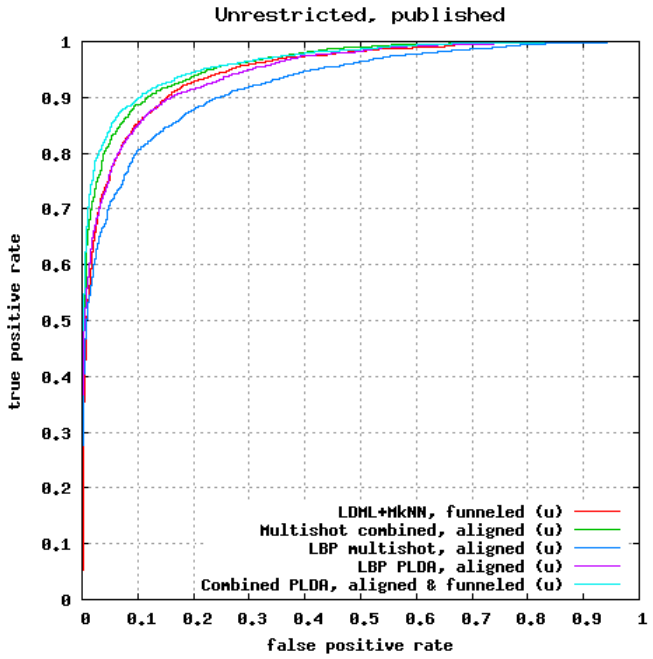


Figure 2.7: ROC curves on LFW for methods trained using the unrestricted protocol.

accuracy on many common face databases [84]. Despite the progress that has been made, there still exists a significant gap between machine-level performance and human-level performance. Existing work has pointed to variations in pose as one of the most challenging aspects of unconstrained face verification, and the source of many of the errors made on LFW [37, 83].

The success of LFW has inspired several similar databases focusing on verification in unconstrained environments. The Public Figures Face Database (Pubfig)[13] is also a benchmark for unconstrained face verification, but with an emphasis on containing more images of each person (with a smaller total number of persons in the database) [49]. The Action Similarity Labeling (ASLAN) Challenge[14] is a database for benchmarking performance of action recognition in videos taken from YouTube [48].

[13] http://www.cs.columbia.edu/CAVE/databases/pubfig/

[14] http://www.openu.ac.il/home/hassner/data/ASLAN/ASLAN.html

CHAPTER 3

UNSUPERVISED JOINT ALIGNMENT

Many recognition algorithms depend on careful positioning of an object into a canonical pose, so the position of features relative to a fixed coordinate system can be examined. This positioning is generally done either manually or by training a class-specialized learning algorithm with samples of the class that have been hand-labeled with parts or poses. In this chapter, we describe a novel method to achieve this positioning using poorly aligned examples of a class with no additional labeling. Given a set of unaligned exemplars of a class, such as faces, we *automatically* build an alignment mechanism, without any additional labeling of parts or poses in the data set. Using this alignment mechanism, new members of the class, such as faces resulting from a face detector, can be precisely aligned for the recognition process. Our alignment method improves performance on a face recognition task, both over unaligned images and over images aligned with a face alignment algorithm specifically developed for and trained on hand-labeled face images [35]. We also demonstrate its use on an entirely different class of objects (cars), again without providing any information about parts or pose to the learning algorithm.

3.1 Introduction

The identification of certain objects classes, such as faces or cars, can be dramatically improved by first transforming a detected object into a canonical pose. Such registration reduces the variability that an identification system or classifier must contend with in the modeling process. Subsequent identification can condition on spatial

position for a detailed analysis of the structure of the object in question. Thus, many recognition algorithms assume the prior rough alignment of objects to a canonical pose [3, 41, 65, 106]. In general, the better this alignment is, the better identification results will be. In fact, alignment itself has emerged as an important sub-problem in the face recognition literature [109], and a number of systems exist for the detailed alignment of specific categories of objects, such as faces [6, 14, 34, 39, 59, 118, 119].

The effect of alignment on face recognition can be seen in Figure 3.1. A common approach to determining if the two images presented in the top row are of the same person or not is to extract patches from the same location in each image and test the patch-level similarity. Due to differences in head pose, facial features appear in different locations in each image, and therefore the image patches will have large dissimilarity despite both images being of the same person. The red circles indicate eyes and nose position in the left image, and are not present in the original images. In this chapter, we present an unsupervised method that automatically aligns the images, producing the bottom row of images. Alignment removes the undesired variability due to in-plane rotation and places the facial features into close correspondence.

Previous work on face image alignment has focused on the supervised approach of Active Appearance Models [14] and its extensions, such as Active Wavelet Networks [34], Bayesian Mixture Models [119], Direct Appearance Models [59], variable illumination [39], and Bayesian Tangent Shape Models [118]. These methods require a set of training images to be manually labeled with corresponding landmarks, typically around 600 training images with 80 landmarks, such as in [118].

A somewhat different method for face alignment is given by Berg *et al*. [6], which uses support vector machines to detect specific facial features, such as corners of eyes and tip of nose. The SVMs are trained from 150 hand labeled faces, then the output of the SVMs on new images is used to align the images to a canonical pose. While this method works well for a subset of the images in their data set, they throw out

Figure 3.1: Top: A pair of images from the Labeled Faces in the Wild database, where the objective is to determine if both images are of the same person or two different people. Bottom: The same pair of images, after unsupervised alignment. In this instance, unwanted variability due to in-plane rotation is removed, placing facial features in both images into the same image location and allowing for more accurate face recognition. Red circles indicate eyes and nose position in the left image, and are not present in the original images.

images with low alignment score, eliminating over 20 percent of their training data. Examples of poor alignment results from Berg are shown in Figure 3.2.

Discarding bad alignments is appropriate for their application, where the goal is to cluster images with similar identity. However, our goal is to produce better alignments for every image, for example to align images to improve recognition, and for such applications one cannot discard difficult to align images.

We point out that it is frequently much easier to obtain images that are roughly aligned than those that are precisely aligned, indicating an important role for automatic alignment procedures. For example, images of people can be easily acquired using a camera in an indoor environment triggered by a motion detector. However, the resulting images will not be precisely aligned.

Although there exist many individual components to do both detection and recognition, we believe one of the most significant obstacles to the creation of a complete

Figure 3.2: Examples of poor alignment using method of Berg *et al*.

end-to-end system capable of performing recognition from an arbitrary scene is in the difficulty of alignment, the middle stage of the recognition pipeline (Figure 2.2 in Chapter 2). Often, the middle stage is ignored, with the assumption that the detector will perform a rough alignment, leading to suboptimal recognition performance.

A system that did attempt to address the middle stage would suffer from two significant drawbacks of current alignment methods:

- They are typically designed or trained for a single class of objects, such as faces.
- They require the manual labeling either of specific features of an object (like the middle of the eye or the corners of the mouth),[1] or a description of the pose (such as orientation and position information).

As a result, these methods require significant additional effort when applied to a new class of objects. Either they must be redesigned from scratch, or a new data set must be collected, identifying specific parts or poses of the new data set before an alignment system can be built. In contrast, systems for the detection and recognition steps of the recognition pipeline only require simple, discrete labels, such as object

[1]Some systems identify more than 80 landmarks per face for 200 to 600 faces [39, 118].

versus non-object or pair match versus pair non-match, which are more straightforward to obtain, making these systems significantly easier to set up than current systems for alignment, where even the form of the supervised input is very often class-dependent.

Some previous work has used detectors capable of returning some information about object rotation, in addition to position and scale, such as, for faces, [45, 95]. Using the detected rotation angle, along with the scale and position of the detected region, one could place each detected object into a canonical pose. However, so far, these efforts have only provided very rough alignment due to the lack of precision in estimating the pose parameters. For example, in [45], the rotation is only estimated to within 30 degrees, so that one of 12 rotation-specific detectors can be used. Moreover, even in the case of frontal faces, position and scale are only roughly estimated, and, in fact, for face images, we use this as a starting point and show that a more precise alignment can be obtained.

More concretely, in this chapter, we describe a system that, given a collection of images from a particular class, automatically generates an "alignment machine" for that object class. The alignment machine, which we call an *image funnel*, takes as input a poorly aligned example of the class and returns a well-aligned version of the example. The system is fully automatic in that it is not necessary to label parts of the objects or identify their initial poses, or even specify what constitutes an aligned image through an explicitly labeled canonical pose, although it is important that the objects be roughly aligned to begin with. For example, our system can take a set of images as output by the Viola-Jones face detector, and return an image funnel which significantly improves the subsequent alignment of facial images.

(We note that the term *alignment* has a special meaning in the face recognition community, where it is often used to refer to the localization of specific facial features. Here, because we are using images from a variety of different classes, we use the term

alignment to refer to the rectification of a set of objects that places the objects into the same canonical pose. The purpose of our alignments is not to identify parts of objects, but rather to improve positioning for subsequent processing, such as an identification task.)

3.1.1 Previous Work

The problem of automatic alignment from a set of exemplars has been addressed previously by Learned-Miller's *congealing* procedure [51]. Congealing as traditionally described works directly on the pixel values in each image, minimizing the entropy of each column of pixels (a pixel stack) through the data set. This procedure works well when the main source of variability in a pixel value is due to misregistration. Congealing has proven to work well on simple binary handwritten digits [63] and magnetic resonance image volumes [52, 121], as well as on curve data [62]. These data sets are free of many of the most vexing types of noise in images. In particular, the goal of this work was to extend congealing-style methods to handle real-world image complexity, including phenomena such as

- complex and variable lighting effects,
- occlusions,
- highly varied foreground objects (for example, for faces, arising from varying head shape, hair, beards, glasses, hats, and so forth), and
- highly varied backgrounds.

For example, on a realistic set of face images taken from news photographs, straightforward implementations of congealing did not work at all. To make the general approach of congealing work on this type of complex images, we needed to define features for congealing that ignore unimportant variability, such as lighting; have a large capture range; and are not sensitive to the clustering procedure we use

to obtain the first two properties. The details of the extension are developed in Section 3.3.

Another information theoretic method was previously proposed by Kim *et al.* [47]. However, that method solves the separate problem of computing correspondences between two highly similar images taken from a stereo pair using mutual information, whereas our method jointly aligns an entire set of highly variable images using entropy minimization.

We demonstrate our system on different classes of images: frontal faces and rear views of cars. For faces, we show high quality results on the Faces in the Wild data set [5], which contains many different people under different poses and lighting, on top of complex backgrounds, in contrast to the data sets on which many other alignment methods are tested, which contain a limited number of people in front of controlled backgrounds. We then show similar quality alignment results on cars, using the same out-of-the-box code as used for the faces, without the need for any training or labeling.

In addition, we do detailed comparisons of our results in frontal face rectification with previous work by Zhou *et al.* [118]. In particular, we show that face identifiers built using our rectified images outperform an identifier built using images that either have not been pre-processed and even exceeds an identifier built from images aligned using Zhou's supervised alignment method.

3.2 Congealing

We first review the basics of congealing. Additional details can be found in Miller [64]. In Section 3.3 we show how to extend this framework to handle complex images.

3.2.1 Distribution Field

A key concept in congealing is the **distribution field**. Let $\mathcal{X} = \{1, 2, \ldots, M\}$ be the set containing all possible feature values at a given pixel. For example, using intensity values as features, for a binary image, $M = 2$, and for a grayscale image, $M = 256$. A distribution field is a distribution over $\mathcal{X}$ at each pixel, so for a binary feature, a distribution field would be a distribution over $\{0, 1\}$ at each pixel in the image.

One can view the distribution field as a generative independent pixel model of images by placing a random variable X_i at each pixel location i. An image then consists of a draw from the alphabet $\mathcal{X}$ for each X_i according to the distribution over $\mathcal{X}$ at the ith pixel of the distribution field.

Another important concept in congealing is the **pixel stack**, which consists of the set of values with domain $\mathcal{X}$ at a specific pixel location across a set of images. Thus, the empirical distribution at a given pixel of a distribution field is determined by the pixel stack at that pixel location.

Congealing proceeds by iteratively computing the empirical distribution defined by a set of images, then for each image, choosing a transformation (for example, over the set of affine transformations) that reduces the entropy of the distribution field. An important point is that, under an independent pixel model and uniform distribution over transformations, minimizing the entropy of the distribution field is equivalent to maximizing the likelihood according to the distribution field [51].

Therefore, an equivalent formulation of congealing is the following: compute the empirical distribution field of a set of images, find the transformation for each image that increases the likelihood of the image under the transformation according to the distribution field, then recalculate the distribution field according to the transformed images, and iterate until convergence.

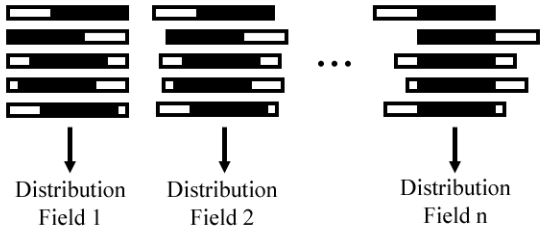


Figure 3.3: Schematic illustration of congealing of one dimensional binary images, where the transformation space is left-right translation

3.2.2 Image Funnel

Once congealing has been performed on a set of images, for example a training set for a face recognition algorithm, there is the question of how to align additional images, such as from a new test set. Theoretically, one could align new images by inserting them into the training set and re-running the congealing algorithm on all the images, but a more efficient technique can be used by keeping the distribution fields produced at each iteration of congealing [51].

By maintaining the sequence of distribution fields from each iteration of congealing, one can align a new image by transforming it, at each iteration, according to the saved distribution field from the corresponding iteration of the original congealing. The sequence of distribution fields begins at higher entropy as the images are initially unaligned, and decreases in entropy as the images are iteratively aligned during congealing. When aligning a new image according to this sequence of distribution fields, the image is sharpened from the initial "wide" distribution to the final "narrow" distribution, and for this reason we refer to the learned sequence of distribution fields of the training congealing as an **image funnel**, and we will refer to the alignment of a new image according to the image funnel as **funneling** to distinguish it from the original congealing.

Figure 3.3 illustrates the process of congealing on one dimensional binary images. At each iteration, the distribution field is a function of the set of transformed images, and the sequence of distribution fields forms an image funnel that can be later used to align new images.

3.3 Methodology

3.3.1 Congealing with SIFT descriptors

We now describe how we have adapted the basic congealing algorithm to work on realistic sets of images. We consider a sequence of possible choices for the alphabet $\mathcal{X}$ on which to congeal. In particular, we discuss how each choice improves upon the previous choice, eventually leading to an appropriate feature choice for congealing on complex images.

In applying congealing to complicated images such as faces from news photographs, a natural first attempt is to set the alphabet $\mathcal{X}$ over the possible color values at each pixel. However, the high variation present in color in the foreground object as well as the variation due to lighting will cause the distribution field to have high entropy even under a proper alignment, violating one of the necessary conditions for congealing to work.

Rather than considering color, one could set $\mathcal{X}$ to be binary, corresponding to the absence or presence of an edge at that pixel. However, another necessary condition for congealing to work is that there must be a "basin of attraction" at each point in the parameter space toward a low entropy distribution.

For example, consider two binary images a and b of the number 1, identical except for an x-translation. When searching over possible transformations to align b to a, unless the considered transformation is close enough to the exact displacement to cause b and a to overlap, the transformation will not cause any change in the entropy of the resulting distribution field.

Another way of viewing the problem is that, when $\mathcal{X}$ is over edge values, there will be plateaus in the objective function that congealing is minimizing, corresponding to neighborhoods of transformations that do not cause changes in the amount of edge overlap between images, creating many zero-gradient problems in the optimization.

Therefore, rather than simply taking the edge values, instead, to generate a basin of attraction, one could integrate the edge values over a window for each pixel. To do this, we calculate the SIFT descriptor [61] over an 8x8 window for each pixel. This gives the desired property, since if a section of one pixel's window shares similar structure with a section of another pixel's window (need not be the corresponding section), then the SIFT descriptors will also be similar. In addition, using the SIFT descriptor gives additional robustness to lighting.

Congealing directly with the SIFT descriptors has its own difficulties, as each SIFT descriptor is a 32 dimensional vector in our implementation, which is too large of a space to estimate entropy without an extremely large amount of data. Instead, we compute the SIFT descriptors for each pixel of each image in the set, and then cluster these using k-means to produce a small set of clusters (in our experiments, we have been using 12 clusters), and let $\mathcal{X}$ be over the possible clusters. In other words, the distribution fields consist of distributions over the possible clusters at each pixel.

After clustering, rather than assigning a cluster for each pixel, we instead do a soft assignment of cluster values for each pixel. Congealing with hard assignments of pixels to clusters would force each pixel to take one of a small number of cluster values, leading to local plateaus in the optimization landscape. For example, in the simplest case, doing a hard assignment with two clusters would lead to the same zero-gradient problems as discussed before with edge values.

This problem of zero-gradients was borne out by preliminary experiments we ran using hard cluster assignments, where we found that the congealing algorithm would

terminate early without significantly altering the initial alignment of any of the images.

To get around this problem, we model the pixel's SIFT descriptors as being generated from a mixture of Gaussians model, with one Gaussian centered at each cluster center and σ_i's for each cluster that maximize the likelihood of the labeling. Then, for each pixel, we have a multinomial distribution with size equal to the number of clusters, where the probability of an outcome i is equal to the probability that the pixel belongs to cluster i. So, instead of having an intensity value at each pixel, as in traditional congealing, we have a vector of probabilities at each pixel.

The idea of treating each pixel as a mixture of clusters is motivated by the analogy to gray pixels in the binary image case. In the binary image case, a gray pixel is interpreted as being a mixture of underlying black and white "subpixels" [51]. In the same way, rather than doing a hard assignment of a pixel to one cluster, we treat each pixel as being a mixture of the underlying clusters.

3.3.2 Implementation

Following the notation in [51], suppose we have N face images, each with P pixels. Let x_i^j be the multinomial distribution of the ith pixel in the jth image, $x_i^j(k)$ be the probability of the kth element of the multinomial distribution in x_i^j, and let $x_i^{j'}$ be the multinomial distribution of the ith pixel of the jth image under some transformation U^j. Denote the pixel stack $\{x_i^{1'}, x_i^{2'}, \ldots x_i^{N'}\}$ as x_i'.

In our congealing algorithm, we first compute the empirical distribution field defined by the images under a particular set of transformations. Define $D_i(k)$ as the probability of the kth element in the distribution at the ith pixel of the distribution field. Then, $D_i(k) = \frac{1}{N}\sum_j x_i^{j'}(k)$. The entropy of a distribution at a particular pixel i is equal to

$$H(D_i) = -\sum_k D_i(k) \log_2 D_i(k). \tag{3.1}$$

Thus, at each iteration in congealing, we wish to minimize the total entropy of the distribution field $\sum_{i=1}^{P} H(D_i)$. This is equivalent to finding, for each image, the transformation that maximizes the log-likelihood of the image with respect to the distribution field, *e.g.* the transformation that maximizes

$$\sum_{i=1}^{P} \sum_k x_i^{j'}(k) \log D_i(k) \tag{3.2}$$

for a given image j. In our case, this maximization is done over the transformations defined by the four parameters, x-translation, y-translation, rotation, and scaling (uniform in x and y), for each image. In our implementation, we do a hill climbing step at each iteration that increases the likelihood with respect to the distribution field at that iteration.

3.4 Experimental Results

In this section, we show experimental results of aligning two object classes, faces and cars, and demonstrate accuracy improvement in the subsequent recognition of faces due to improved alignment.

3.4.1 Alignment on Faces in the Wild

We ran our alignment algorithm on 300 faces selected randomly from the first 300 clusters of the Faces in the Wild data set [5] (the predecessor to LFW). This data set consists of news photographs that cover a wide variety of pose, illumination, and background. We used the Viola-Jones face detector to extract the faces from the images, and ran the images through the congealing alignment algorithm. A representative sample of 75 of the resulting aligned images after congealing are given in Figures 3.6, 3.7, and 3.8. Also shown are the original images, together with the corresponding bounding boxes of the final alignments.

For comparison, we aligned the same set of images using the Zhou face alignment [118] using their web interface,[2] which returns the alignment as a set of connected landmark points. The results are also presented in the alignment samples, and one can see that the two alignment methods are comparable, despite congealing being unsupervised. Both methods do a good job of finding the correct scale of the face, though in a few instances the Zhou alignment is thrown off, such as by partial occlusion due to a tennis racquet or confusing the bottom of the lip as the chin. Both methods also do a good job with respect to rotation, as is most evident in the first picture of the sixth row.

3.4.2 Cars

We also show results on a separate data set of 125 rear car images, taken from different parking lots with variable background and lighting. Since our algorithm is fully automatic, we were able to obtain these results using the same code as with faces without any labeling or training. A representative sample of the final alignment bounding boxes are given in Figure 3.5. Of the 50 images, only one is a clear error (6th row, 2nd column), and one is a case where the algorithm rotated the image in the right direction but not to a sufficient degree (7th row, 4th column). Of the other 75 images, the final bounding box captures the correct scale, rotation, and position of the car, with the exception of one other car where the algorithm again rotated the image in the right direction but not sufficiently. We emphasize again that no changes of any kind were made to the code before running the car examples; the algorithm ran directly as it did on the faces. We believe this is a dramatic demonstration of the generality of this method.

---

[2] `http://facealignment.ius.cs.cmu.edu/alignment/webdemo.html`

3.4.3 Improvement in Recognition

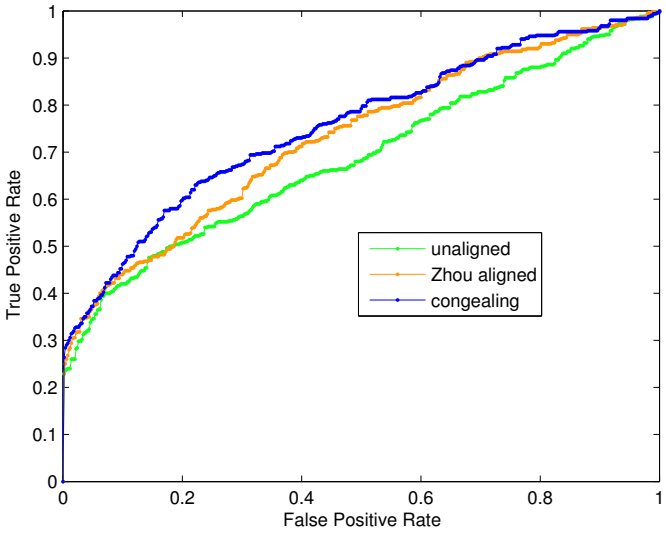
In addition, we also tested the performance of a face recognizer on three different alignment processes. We used a hyper-feature based recognizer of Jain *et al.* [41] with 500 randomly selected training pairs and 500 randomly selected test pairs from the Faces in the Wild data set.

For the baseline of our comparison, we trained and tested the recognizer with the unaligned face images found by the Viola-Jones face detector. Next, we examined how aligning the face images with the Zhou method and with congealing would affect the results. We used the unaligned images from the Viola-Jones face detector as input into the two systems, which, for each image, produce a similarity transformation used to align that particular image. For the congealing alignment, we aligned the images by funneling the output of the Viola-Jones face detector using the image funnel learned from congealing on the 300 faces above.

We chose to compare against the Zhou alignment algorithm rather than the Berg method presented in [6]. The Berg algorithm uses support vector machines to detect specific facial features, such as corners of eyes and tip of nose, that are then used to align the images to a canonical pose. Although this method works well for a subset of the images in their data, they throw out images with low alignment score, eliminating a large number of faces. While discarding bad alignments is appropriate for their application, for the purpose of recognition, one cannot discard difficult to align images.

On the other hand, the Zhou system is designed for detection and face point localization in addition to pose estimation, and not specifically to improve classification accuracy. However, it is reasonable to adopt the system for the purposes of alignment to a fixed coordinate system and seemed to align faces as well as anything else we found. We took care to make the comparison fair (by using the default unaligned

image when no face was detected by the Zhou system and by manually picking the best face when the Zhou system detected multiple faces for a given image).



	unaligned	Zhou aligned	congealing
AUC	0.6870	0.7312	0.7549

Figure 3.4: ROC curves and area under curves for recognition. Using face images aligned with congealing during both training and testing of a face identifier uniformly improves accuracy, not only over images directly from the Viola-Jones detector ("unaligned") but also on images that have been aligned using the method of Zhou *et al*.

The ROC curves for the recognition, as well as the area under the curves, are given in Figure 3.4. From this figure, it is clear that our method, which is completely automated and requires no labeling of pose or parts, substantially improves the results of recognition over the outputs of the Viola-Jones face detector, and even exceeds the supervised alignment method of Zhou in performance benefit to recognition.

3.5 Discussion

In this chapter, we have presented an unsupervised technique for jointly aligning images under complex backgrounds, lighting, and foreground appearance. Our

method obviates hand-labeling hundreds of images while maintaining comparable performance with supervised techniques. In addition, our method increases the performance of a face recognizer by precisely aligning the images. Of course, our method is not completely unsupervised in the sense that it must be provided with images of objects of a particular class. However, in many scenarios, such images can be automatically acquired, especially since detailed alignment is not a requirement.

One possible extension of our method is to align images in a two part process: First, all the images are aligned using congealing, then the quality of the alignment is estimated for each image so that poorly aligned images can be re-aligned in a separate second stage. The quality of the alignment could be estimated from the likelihood of each image under its alignment according to the final distribution field.

Another possible extension is to use the multi-view face detector in [45] to first separate face images into three separate categories: frontal, left profile, and right profile, and then attempt to align each category of faces individually.

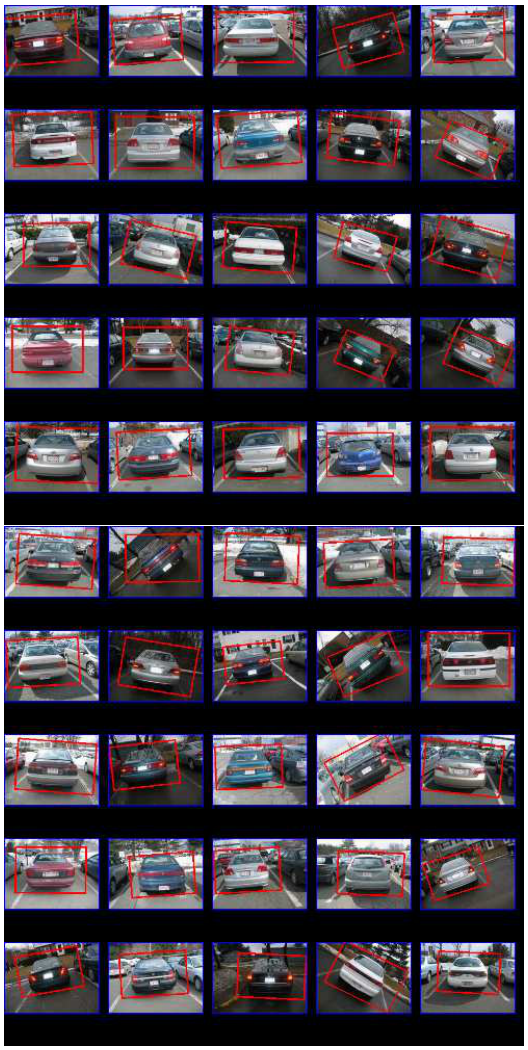


Figure 3.5: Input to congealing with bounding boxes of final alignment

Figure 3.6: A sample of aligned images: The left column shows the aligned images as output by congealing. The middle column shows the original images as input to congealing, with bounding boxes determined from final alignment. The right column shows the results of the Zhou alignment.

Figure 3.7: A sample of aligned images: The left column shows the aligned images as output by congealing. The middle column shows the original images as input to congealing, with bounding boxes determined from final alignment. The right column shows the results of the Zhou alignment.

Figure 3.8: A sample of aligned images: The left column shows the aligned images as output by congealing. The middle column shows the original images as input to congealing, with bounding boxes determined from final alignment. The right column shows the results of the Zhou alignment.

CHAPTER 4

DEEP LEARNING FOR FACE VERIFICATION

Most modern face recognition systems rely on a feature representation given by a hand-crafted image descriptor, such as Local Binary Patterns (LBP) [76], and achieve improved performance by combining several such representations. In this chapter, we propose deep learning as a natural source for obtaining additional, complementary representations.

To learn features in high resolution images, we make use of convolutional deep belief networks [55]. Moreover, to take advantage of global structure in an object class, we develop local convolutional RBMs, a novel extension of convolutional models that make use of this structure by not assuming stationarity of features across the image, while maintaining scalability and robustness to small misalignments. We also present a novel application of deep learning to representations other than pixel intensity values, such as LBP. We compare performance of networks trained using unsupervised learning against networks with random filters, and show empirically that learning weights is necessary for obtaining good multi-layer representations, and additionally provides robustness to the choice of network architecture parameters.

We show that a recognition system using only representations obtained from deep learning can achieve comparable accuracy with a system using a combination of hand-crafted image descriptors. By further combining the two representations, we can achieve state of the art results on LFW.

4.1 Introduction

There has been a significant amount of progress made in the area of face recognition, with recent research focusing on the face verification (pair matching) problem. As described in more detail in Chapter 2, in this set-up, pairs of images are given at training time, along with a label indicating whether the pair contains two images of the same person (matched pair), or two images of two different persons (mismatched pair). At test time, a new pair of images is presented, and the task is to assign the appropriate matched/mismatched label. Unlike other face recognition problem formulations, it is not assumed that the person identities in the training and test sets have any overlap, and often the two sets are disjoint.

This set-up removes one of the fundamental assumptions of the traditional experimental design, making it possible to perform recognition on never-before-seen faces. Another important assumption that has been relaxed recently is the amount of control the experimenter has over the acquisition of the images. In unconstrained face verification, the only assumption made is that the face images were detected by a standard face detector. In particular, images contain significant variations in nuisance factors such as complex background, lighting, pose, and occlusions. These factors lead to large intra-class differences, making the unconstrained face verification problem very difficult.

The current standard for benchmarking performance on unconstrained face verification is the Labeled Faces in the Wild (LFW) data set presented in Chapter 2. Since the release of the database, classification accuracy on LFW has improved dramatically, from initial methods getting less than 0.75 accuracy to current state-of-the-art methods getting 0.84 to 0.86 accuracy [112].

The majority of existing methods for face verification rely on feature representations given by hand-crafted image descriptors, such as SIFT [61] and Local Binary Patterns (LBP) [76]. Further performance increases are obtained by combining sev-

eral of these descriptors [112]. Rather than spending time attempting to engineer new image descriptors by hand, we instead propose obtaining new representations automatically through unsupervised feature learning with deep network architectures [31, 4, 90, 87, 50].

These representations offer several advantages over those obtained through hand-crafted descriptors. They can capture higher order statistics such as corners and contours, and can be tuned to the statistics of the specific object classes being considered (*e.g.*, faces). An end system making use of deep learning features can be more readily adapted to new domains where the hand-crafted descriptors may not be appropriate.

The primary contributions made in this chapter are:

1. We develop local convolutional RBMs, a novel extension of convolutional RBMs that are able to adapt to the global structure in an object class, while still being able to scale to high resolutional images and be robust to minor misalignment.

2. We present a novel application of deep learning to a Local Binary Pattern representation rather than pixel intensity representation, demonstrating the potential to learn additional representations that capture higher order statistics of hand-crafted image descriptors.

3. We evaluate the role of learning in deep convolutional architectures, and find that although random filters perform surprisingly well for single layer models (consistent with work such as [98]), learning filters is necessary to obtain useful multi-layer networks, and also helps in being more robust to the choice of architectural hyperparameters.

4. We demonstrate that, despite the amount of effort spent engineering good image descriptors, by using representations obtained from deep learning, we are able

to achieve comparable accuracy with state of the art methods using these hand-crafted descriptors. Moreover, the information captured by the deep learning representations is complementary to the hand-crafted descriptors, and by combining the two sets of representations, we are able to improve the state of the art face verification results on LFW.

4.2 Background

Here we review relevant work on unconstrained face verification and on deep belief networks for feature representation.

4.2.1 Unconstrained Face Verification

As mentioned in the introduction, the top performing face recognition systems generally use some number of hand-crafted image descriptors such as LBP. Cao *et al.* [11] form a pixel-level feature representation by circular sampling similar to LBP, then quantize these feature vectors using random-projection trees. Classification is done using multiple representations and comparing L_2 distance.

Wolf *et al.* [112] use a "One-Shot Similarity" (OSS) measure and extensions such as "Two-Shot Similarity" (TSS). The idea of OSS is to learn a discriminative model specific to a pair of test images by using a set of background samples. A model is learned that separates one image in the pair from the background images, and is then applied to classify the other image in the pair, and this is repeated for the other image. By combining OSS and TSS using both linear discriminant analysis (LDA) and support vector machines (SVM), over variants of LBP and SIFT descriptors, this method has set the current state-of-the-art results on LFW.

Nguyen and Bai [71] apply cosine similarity learning metric (CSML) to face verification, combining pixel intensity, LBP, and Gabor representations. As this approach

achieves high accuracy using a small number of representations compared with [112], we use a variation on this method in our work, which we describe in Section 4.3.

Kumar *et al.* [49] take a different approach, using additional outside supervised training data to learn binary classifiers for attributes such as gender, goatee, and round face, and binary classifiers that recognize a particular facial region of a particular person, referred to as simile classifiers. Face images are represented as vectors over the outputs of these different classifiers, and classification is performed using an SVM with a radial basis function kernel.

Deep learning has also been previously applied to face verification, and we describe this method in the next section. Pinto and Cox [82] also make use of a multi-layer architecture, where, rather than learning filters, they perform high-throughput screening by employing high-end graphics hardware and performing brute-force search for good feature representations.

Yin *et al.* [114] leverage pose information from the Multi-PIE face database [28], in the form of images of the same face taken from a number of known poses, and apply this information to handle intra-class variation in LFW. By attempting to correct for intra-personal variation, they achieve state of the art performance, for methods that make use of labeled training data external to LFW.

4.2.2 DBNs and Learning

A deep belief network (DBN) is a generative graphical model consisting of a layer of visible units and multiple layers of hidden units, where each layer encodes correlations in the units in the layer below [31]. DBNs and related unsupervised learning algorithms such as auto-encoders [4] and sparse coding [77, 53] have been used to learn higher-level feature representations from unlabeled data, suitable for use in tasks such as classification. These methods have been successfully applied to

computer vision tasks [89, 117, 55, 113, 88, 43], as well as audio recognition [56, 66], natural language processing [13], and information retrieval [96].

Nair and Hinton [69] applied deep learning to object recognition and face verification, using a modification to binomial units that they refer to as noisy rectified linear units. To make learning computationally tractable, they subsample the face images to 32x32. In addition, their method was not translation invariant and had to rely on manual alignment through hand-corrected eye coordinates as preprocessing. In contrast, we take a convolutional learning approach, thus we are able to train the models directly on the full-sized images without relying on careful manual alignment.

As other related work, Ranzato *et al.* [91] proposed a deep generative model with applications to face recognition (e.g., classification). Also, Susskind *et al.* [103] applied 3-way RBMs for modeling pairs of face images. Compared to these models, we consider more scalable algorithms that can be applied larger-sized images (150x150 pixels vs. 48x48 pixels) and focus on the challenging task of face verification.

Our work also studies three different strategies for training the deep learning architecture. The straightforward approach is to train the model using images drawn from the same distribution as the distribution the test images are drawn from, which in our case would be learning from faces in the training set. In many machine learning problems, however, we are given only a limited amount of labeled data, and this can cause an overfitting problem. Thus, we also examine the strategy of self-taught learning [86] (related to semi-supervised learning [72, 12] and transfer learning [105]). The idea of self-taught learning is to use a large amount of unlabeled data from a generative distribution that is different from that of the labeled data, and "transfer" low-level structures that can be shared between unlabeled and labeled data. For instance, we can imagine, for a binary image classification problem of classifying cars versus motorcycles, using a virtually unlimited amount of unlabeled images that can be cheaply obtained through the web.

In the case of generic object categorization tasks, Raina *et al.* [86] and Lee *et al.* [55] have shown successful applications of self-taught learning, using sparse coding and deep belief networks to learn feature representations from natural images. However, self-taught learning has not been used for face verification tasks.

Unlike categorizing generic object images, face verification focuses on a much more restricted subset of images (*i.e.*, faces), requiring a fine granularity of discrimination solely between images within this restricted class. Therefore, there are two interesting questions: first, whether features learned from faces, which have been trained to be useful for generating face images, are useful for discriminating between different faces; and second, whether features obtained from self-taught learning capture useful structures and representations that can be "transferred" from natural images to the face verification problem.

In addition, recent work has shown that random filters can give good performance in a convolutional architecture [98]. This has led to the suggestion that one test different architectures quickly using random filters, and then select the top performing architecture to use with learned weights. In this chapter, we evaluate this strategy for the task of face verification using a multiple-layer deep architecture.

4.3 Methods

In this section, we describe the face verification algorithm we use and the deep learning architectures we apply to learn representations for the verification algorithm.

4.3.1 Recognition Algorithm

Our face verification algorithm is a metric-learning approach inspired by Cosine Similarity Metric Learning (CSML) [71]. For the hand-crafted model, we use the same features as in CSML (pixel intensity, LBP, Gabor). For all feature representations, we use PCA to reduce the dimensionality to 500. We then apply Information-Theoretic

Metric Learning (ITML) [17] to produce a Mahalanobix matrix M, and then perform a Cholesky decomposition yielding a matrix A such that $A'A = M$.

Letting x be the representation of an image after applying PCA, we obtain a feature vector for an image by unit-normalizing Ax. We then form a feature vector for a pair of images by combining the image feature vectors using element-wise multiplication. Finally, we apply a linear SVM to the feature vectors for pairs of images to perform face verification.

In practice, we find that using ITML improves performance over CSML by several percentage points. Note that if A is the identity matrix and the weights of the SVM are 1, then our system reduces to cosine similarity. Consistent with previous work [11], we found that compression using PCA followed by normalization gave the best performance.

4.3.2 Deep Learning

We first review the convolutional restricted Boltzmann machine (CRBM) and the convolutional deep belief network (CDBN) [55]. We then present its extension, the *local CRBM*.

4.3.2.1 Convolutional RBM and DBN

The convolutional restricted Boltzmann machine is an extension of the restricted Boltzmann machine (RBM). The RBM is a Markov random field with a hidden layer and a visible layer (corresponding to image pixels in computer vision problems), with bipartite connections between the layers (*i.e.*, there are no connections among visible nodes or among hidden nodes). In a CRBM, rather than fully connecting the hidden layer and visible layer, the weights between the hidden units and the visible units are local (*i.e.*, 10x10 pixels instead of full image) and shared among all locations in the hidden units. The CRBM captures the intuition that if a certain image feature (or pattern) is useful in some locations of the image, then the same image feature can also

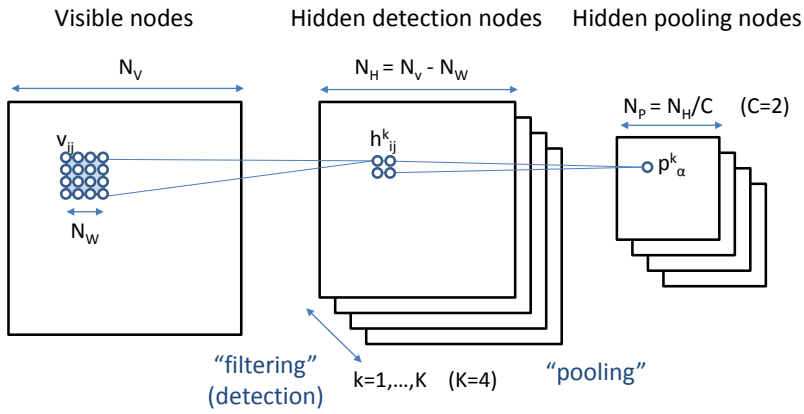


Figure 4.1: Schematic diagram of convolutional RBM with probabilistic max-pooling. For illustration, we used pooling ratio $C = 2$ and number of filters $K = 4$. See text for details.

be useful in other locations. Moreover, by tying the weights between different hidden units, the amount of training data used to estimate a particular weight is increased.

In this chapter, we utilize a convolutional RBM with real-valued visible input nodes $\mathbf{v}$ and binary-valued hidden nodes $\mathbf{h}$. The visible input nodes can be viewed as pixel values in the $N_V \times N_V$ pixel image, and the hidden nodes are organized in 2-D configurations (*i.e.*, $\mathbf{v} \in \mathbb{R}^{N_V \times N_V}$ and $\mathbf{h} \in \{0,1\}^{N_H \times N_H}$).

An illustration of a CRBM can be found in Figure 4.1. The CRBM has three sets of parameters: (1) K convolution filter weights between a hidden node and the visible nodes, where each filter is $N_W \times N_W$ pixels (*i.e.*, $W^k \in \mathbb{R}^{N_W \times N_W}, k = 1, ..., K$); (2) hidden biases $b^k \in \mathbb{R}$ that are shared among hidden nodes; and (3) a visible bias $c \in \mathbb{R}$ that is shared among visible nodes.

To make CRBMs more scalable, Lee *et al*. further developed "probabilistic max-pooling", a technique for incorporating local translation invariance. Max-pooling refers to operations where a local neighborhood (*e.g.*, 2x2 grid) of feature detection outputs is shrunk to a pooling node by computing the maximum of the local neighbors. Max-pooling makes the feature representation become more invariant to local

translations in the input data, and it has been shown to be useful in many computer vision problems [43, 9]. Probabilistic max-pooling enables the CRBM to incorporate max-pooling like behavior, while allowing probabilistic inference (such as bottom-up and top-down inference). It further enables increasingly more invariant representations as we stack CRBMs [27].

We can define the energy function of the probabilistic max-pooling CRBM as follows:

$$\begin{aligned} P(\mathbf{v},\mathbf{h}) &= \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h})) \\ E(\mathbf{v},\mathbf{h}) &= -\sum_{k=1}^{K}\sum_{i,j=1}^{N_H}\sum_{r,s=1}^{N_W} h_{ij}^k W_{rs}^k v_{i+r-1,j+s-1} + \\ &\quad \sum_{i,j=1}^{N_V}\frac{1}{2}v_{ij}^2 - \sum_{k=1}^{K} b_k \sum_{i,j=1}^{N_H} h_{ij}^k - c\sum_{i,j=1}^{N_V} v_{ij} \\ &\text{s.t.} \quad \sum_{(i,j)\in B_\alpha} h_{i,j}^k \le 1, \quad \forall k,\alpha. \end{aligned}$$

Here, B_α refers to a $C \times C$ block of locally neighboring hidden units $h_{i,j}^k$ that are pooled to a pooling node p_α^k.

Under this energy function, the conditional probabilities can be computed as follows:

$$P(v_{ij}=1|\mathbf{h}) = \mathcal{N}((\sum_k W^k *_f h^k)_{ij} + c, 1) \tag{4.1}$$

$$P(h_{i,j}^k=1|\mathbf{v}) = \frac{\exp(I(h_{i,j}^k))}{1+\sum_{(i',j')\in B_\alpha}\exp(I(h_{i',j'}^k))}, \tag{4.2}$$

where $I(h_{ij}^k) \triangleq b_k + (\tilde{W}^k *_v \mathbf{v})_{ij}$, $\mathcal{N}(\cdot)$ is a normal distribution, $\tilde{W}$ refers to flipping the original filter W in both upside-down and left-right directions, $*_v$ denotes valid convolution, and $*_f$ denotes full convolution.[1]

[1]Let $\mathbf{v} \in \mathbb{R}^{N_V \times N_V}$, $\mathbf{h} \in \{0,1\}^{N_H \times N_H}$, and $W^k \in \mathbb{R}^{N_W \times N_W}$, with $N_H = N_V - N_W + 1$. By valid convolution, we mean the region of the convolution that is computed without using any zero-

At the same time, the pooling node p_α^k is a stochastic random variable that is defined as $p_\alpha^k \triangleq \sum_{(i,j)\in B_\alpha} h_{i,j}^k$, and the marginal posterior can be written as a softmax function:

$$P(p_\alpha^k = 1|\mathbf{v}) = \frac{\sum_{(i',j')\in B_\alpha} \exp(I(h_{i',j'}^k)}{1 + \sum_{(i',j')\in B_\alpha} \exp(I(h_{i',j'}^k))}. \tag{4.3}$$

When sampling from the posterior (given the visible nodes), we can efficiently sample the hidden nodes in each block in parallel from multinomial distributions, then set the pooling node values accordingly.

The objective function is the log-likelihood of the training data. Although exact maximum likelihood training is intractable, the contrastive divergence approximation allows us to estimate an approximate gradient efficiently [30]. Contrastive divergence is not unbiased, but has low variance, and has been successfully applied in optimizing many undirected graphical models that have intractable partition functions [94, 110, 31].

As in Lee *et al.*, we also apply sparsity regularization. Since the model is highly over-complete, it is necessary to regularize the model to prevent it from learning trivial or uninteresting feature representations (cf., see [77, 90] for other methods for enforcing sparsity.) Specifically, we add a sparsity penalty term to the log-likelihood objective to encourage each hidden unit group to have a mean activation close to a small constant. We implemented this with the following simple update rule (following each contrastive divergence update):

$$\Delta b_k \propto p - \frac{1}{N_H^2} \sum_{i,j} P(h_{ij}^k = 1|\mathbf{v}), \tag{4.4}$$

---

padding, such that $W^k *_v \mathbf{v}$ produces a result of size $N_H \times N_H$. By full convolution, we mean that zero-padding is used, such that $W^k *_f h^k$ produces a result of size $N_V \times N_V$.

where p is a target sparsity, and each image is treated as a mini-batch (meaning that the CRBM parameters are updated after processing each image). The learning rate for the sparsity updates was chosen to make the hidden group's average activation (over entire training data) close to the target sparsity, while allowing variations of activations depending on specific input images.

Sohn *et al*. [101] showed that the sparse RBM could be seen as a relaxation of an RBM with a softmax constraint (where at most one hidden unit is activated), and further, that an RBM with softmax constraint and Gaussian visible units is equivalent to a Gaussian mixture model. They showed that better results could be obtained by initializing the weights in a sparse RBM using the output of a Gaussian mixture model trained using expectation maximization. We use this same initialization strategy. In Appendix B, we give the details on initializing a sparse RBM with binary visible units using an equivalence to a mixture of Bernoullis model.

After training a max-pooling CRBM, we can use it to compute the posterior of the hidden (pooling) units given the input data. These hidden (pooling) unit "activations" can be used as input to further train the next CRBM layer.

By stacking the CRBMs, the algorithm can capture high-level features, such as hierarchical object-part decompositions. In our experiments, we trained up to the third layer. After constructing a convolutional deep belief network, we perform (approximate) inference of the whole network in a feedforward (bottom-up) manner.

4.3.2.2 Local Convolutional RBM

The weight sharing scheme in a CRBM assumes that the distribution over features is stationary in an image with respect to position. However, for images belonging to a specific object class, such as faces, this assumption is no longer true. One strategy for removing this stationarity assumption is to connect each hidden unit to only a local receptive field in the visible image, as in the CRBM, but remove

the parameter tying between weights for different hidden units [91]. However, even with only local connections, without any parameter tying, it is computationally and statistically intractable to scale this model to high resolution images such as used in LFW, where the full images have 250x250 resolution. Moreover, without parameter tying, the model becomes sensitive to local deformations and misalignments.

To maintain the advantages of a CRBM while taking advantage of global structure, we divide the image into a number of overlapping regions. A *local* convolutional restricted Boltzmann machine extends the CRBM by using a separate set of weights for each region. When trained on images with some global structure, a local CRBM can learn a more efficient representation than a CRBM since features are only learned for a particular position if they are useful for the corresponding region. Moreover, since features are no longer shared globally, a local CRBM may be able to avoid spurious activations of a feature that is only present in a certain location.

We can formulate a local CRBM as follows. First, we divide the image into L overlapping regions, with the l-th region defined as $\{R_l : (r^l_{min}, r^l_{max}, c^l_{min}, c^l_{max})\}$, where r and c represent row or column index for the region in the image. For convenience of presentation, we assume that each region is square, with height and width equal to N_R. We denote by V^l the "submatrix" of the visible units that correspond to the l-th region. Let each region have K filters W^l_k of size $N_w \times N_w$. The hidden units h^l_k are binary random variables with 2D spatial structure, having size $N_H \triangleq N_R - N_W + 1$.

We can now define the energy function of the local convolutional RBM as follows:[2]

$$\begin{aligned} E(v,h) &= -\sum_{l=1}^{L}\sum_{k=1}^{K}\left(V^l * \widetilde{W^l_k}\right) \bigodot H^l_k \\ &\quad + \sum_{ij}\frac{1}{2}(V^l_{ij} - c)^2 + \sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{r=1}^{N_H}\sum_{s=1}^{N_H} b^l_k \left(H^l_k\right)_{r,s}, \end{aligned}$$

[2]Note that we can also define probabilistic max-pooling for the local CRBM. However, for the simplicity of presentation, we present a case without probabilistic max-pooling.

where $\bigodot$ is the element-wise product operator.

Given V fixed, the conditional probability of H can be defined as

$$P(H_k^l|V^l) = \sigma(V^l * \widetilde{W_k^l} + b_k^l),$$

where the $\sigma(x) = \frac{1}{1+\exp(-x)}$. We can also define the conditional probability of the visible units given the hidden units as

$$P(V|H) = \mathcal{N}\left(\sum_l I^l\left(\sum_k W_k^l * H_k^l\right) + c, I\right).$$

$I^l(Y)$ is a projection operator from $R^{N_R \times N_R} \rightarrow R^{N_V \times N_V}$, where Y is an $N_R \times N_R$ image used to accumulate the contribution of each local region to the visible layer. $I^l(Y)$ is defined as

$$\left[I^l(Y_{r',c'})\right]_{r,c} = \begin{cases} Y_{r',c'} & \text{if } (r,c) = (r' + r_{min}^l - 1, c' + c_{min}^l - 1) \\ 0 & \text{otherwise.} \end{cases}$$

With these conditional probabilities, we can train the local CRBM following the same procedure as for the CRBM using contrastive divergence.

4.3.2.3 Learning from Other Representations

Deep learning for images is usually performed by letting the visible units be whitened pixel intensity values. We learn additional novel representations by learning deep networks on Local Binary Patterns, demonstrating the potential for learning representations that capture higher order statistics of hand-crafted image descriptors. Using uniform LBPs (at most two bitwise transitions), we have a 59 dimensional binary vector at each pixel location. We find a small increase in performance by first

forming histograms of 3x3 neighbors (average pooling), and then learning a binary CRBM on this representation.

4.4 Experiments

For our experiments, we used the LFW-a[3] face images aligned using commercial face alignment software, provided in [112].[4] We use three croppings of each image (150x150, 125x75, 100x100), resizing to the same input size for the visible layer, to capture information at different scales. For self-taught learning, we used images from the Kyoto natural images data set [18].[5]

We used the authors' implementation of ITML.[6]. To solve the SVM, we use the Shogun Toolbox [102].[7] We set the SVM C parameter using the development view of LFW. We optimized our CDBN code to use a GPU,[8] allowing us to test a single kernel system in several minutes and learn weights in a DBN in less than an hour.

4.4.1 Setting Architecture and Model Hyperparameters

One of the challenges of using a deep learning architecture is the number of architecture and model hyperparameters that one must set. For a CDBN, we must decide the size of the input image, and for each layer, the size of the filters, number of filters, max-pooling region size, and sparsity of the hidden units when learning the filters.

Saxe *et al*. [98] found some correlation between performance with random filters and learned filters for a given architecture, and suggested using search over archi-

---

[3] `http://www.openu.ac.il/home/hassner/data/lfwa/`

[4] We used LFW-a, as these experiments were carried out prior to the development of the unsupervised alignment method presented in Chapter 5.

[5] `http://www.cnbc.cmu.edu/cplab/data_kyoto.html`

[6] `http://www.cs.utexas.edu/~pjain/itml/`

[7] `http://www.shogun-toolbox.org/`

[8] We used code from Graham Taylor: `http://www.cs.nyu.edu/~gwtaylor/code/GPUmat/`.

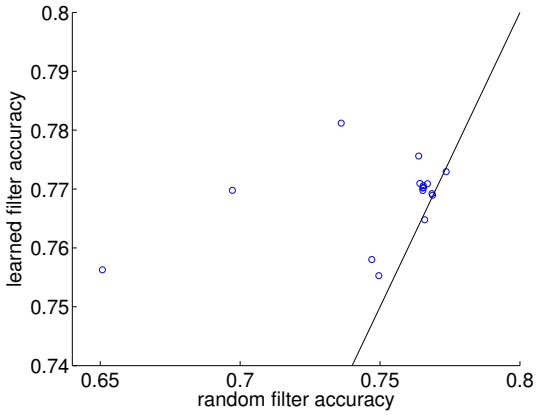


Figure 4.2: Random filter accuracy versus learned filter accuracy. The line indicates the diagonal $y = x$. From this figure, it can be seen that although there is some correlation between random filter accuracy and learned filter accuracy, learning filters has the benefit of being robust to the choice of architecture, increasing the accuracy significantly for architectures where random filters give low accuracy.

tectures with random filters as a proxy for selecting a best architecture to use with learned weights.

We first evaluated the correlation between random weight and learned weight performance for a one layer network with 16 different architectures, varying the above architecture hyperparameters. Figure 4.2 shows a scatter plot of random weight performance versus learned weight performance. We find a somewhat high correlation of 0.40. However, a more interesting finding is that the range of accuracies for the learned filters is much more concentrated around higher values compared with the random filters. Thus, we hypothesize that, while networks with random filters can approach the same accuracy as networks with learned filters, given the right architecture, an added benefit of learning is that the accuracy becomes more robust to the specific architecture hyperparameters.

Moreover, we find that multi-layer networks with random weights at each layer yield representations that lead to near chance recognition performance. Empirically, this seems to indicate that, at least for the face verification task, the non-linearities

Source	Rep.	Layer	Model	Accuracy
Kyoto	Int.	1	CRBM	0.8527
Faces	Int.	1	CRBM	0.8530
Kyoto	Int.	2	CRBM	0.8522
Faces	Int.	2	CRBM	0.8457
Faces	Int.	2	local CRBM	0.8508
Kyoto	LBP	1	CRBM	0.8520
Faces	LBP	1	CRBM	0.8485
Kyoto	Int.	1+2		0.8572
Faces	Int.	1+2		0.8582
Kyoto	both	1+2		0.8660
Faces	both	1+2		0.8642
both	both	1+2		0.8688

Table 4.1: Verification accuracy with different deep learning architectures and training sources. The second column indicates the representation for the visible units, and Int. stands for whitened pixel intensity values. Top: Single representations. Bottom: Combining representations with linear SVM.

in a multi-layer network is such that random filters in a convolutional model do not give good representations, and learning is necessary. Given these findings, we set the hyperparameters by performing a coarse search over the possible values, and learning and evaluating the model on the development view of LFW.

4.4.2 Results

The top section of Table 4.1 gives the accuracy for individual deep architectures. Since we expect the basic image features learned by a single layer CRBM to be largely edge-like features that are shared throughout the image, we apply our local CRBM model only at the second layer. The second layer CRBM and local CRBM have approximately the same size hidden layer representation, but the local CRBM is able to learn more filters since they are specific to each region, and achieves a higher accuracy. Figure 4.3 shows a visualization of the filters learned by the local CRBM. The bottom section of Table 4.1 gives the accuracy when combining the scores from multiple deep architectures using a linear SVM. As the different layers are capturing

complementary information, we are able to achieve higher accuracy by fusing these scores.

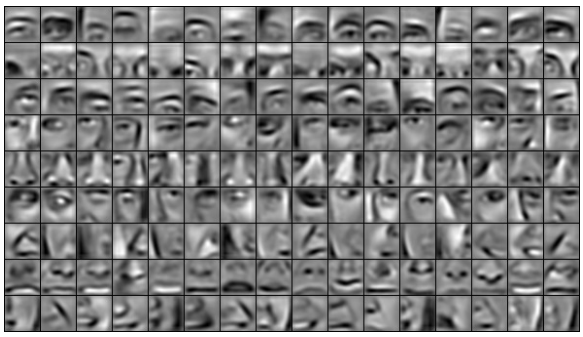


Figure 4.3: Visualization of sample filters from the second layer local CRBM. Each row represent filters corresponding to each local region, where the training images were divided into 9 half-overlapping regions (i.e., the size of each region is half the image size). We can see that the local CRBM capture characteristic facial parts corresponding to the local regions.

Table 4.2 gives the final accuracy of our system using the deep learning representations, and the combined deep learning and hand-crafted image descriptor representations, in comparison with other systems trained using the image-restricted setting of LFW.[9] Our system, using only deep learning representations, is competitive with state of the art methods that rely on a combination of descriptions of hand-crafted image descriptors, and achieves highest accuracy among existing deep learning methods, despite the fact that [69] used manual annotations of eye coordinates to align the faces.

By combining the representations from deep learning and hand-crafted image descriptors, we obtain further improvements and achieve a new state of the art accuracy.

[9]We do not compare with the published accuracies of CSML [71] or High-Throughput Brain-Inspired Features [82], as we believe both methods are using View 1 of LFW in a manner leading to overfitting to View 2, given the overlap between the two views. More information, and a discussion of View 1/View 2 overlap, is presented in Appendix A. In the table, we give the accuracy of CSML using our implementation, following the training strategy presented in the CSML paper, which is not the same as the strategy used to obtain the accuracy numbers in the CSML paper.

Wolf *et al.* [112] combine hand-crafted image descriptors such as LBP, Gabor, and SIFT, and additionally combine each of these representations for six different similarities metrics. Results for a single similarity metric (OSS only) are also given in Table 4.2. Our general methodology of learning additional representations through deep learning could also be applied to multiple similarity metrics rather than just a single metric, potentially further improving our results.

Similarly, the recent paper of Yin *et al.* [114], who achieve state of the art accuracy using external training data containing pose information to handle intra-personal variation, relies on a fusion of four different hand-crafted image descriptors, and could also potentially be improved by adding additional deep learning representations.

Method	$\hat{\mu} \pm S_E$
V1-like with MKL [85]	0.7935 ± 0.0055
Linear rectified units [69]	0.8073 ± 0.0134
CSML [71]	0.8418 ± 0.0048
Learning-based descriptor [11]	0.8445 ± 0.0046
Attribute and simile [49]	0.8529 ± 0.0123
OSS and TSS [112]	0.8683 ± 0.0034
OSS only [112]	0.8207 ± 0.0041
Hand-crafted	0.8718 ± 0.0049
Deep Learning	0.8688 ± 0.0062
Combined	0.8777 ± 0.0062

Table 4.2: Comparison of our method with current state-of-the-art methods on LFW. The right column gives mean classification accuracy and standard error of the mean.

4.5 Analysis

We can gain additional insight into the face verification problem by looking at the number of representations whose score correctly classifies each pair. Figure 4.4 give a histogram over these values, separately for mismatched pairs and matched pairs. Interestingly, the pairs that are correctly classified by few or no representations are heavily skewed toward matched pairs.

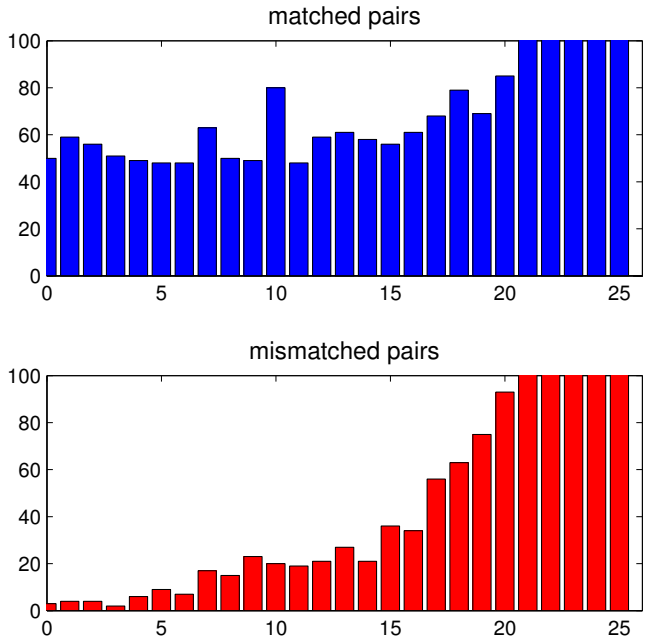


Figure 4.4: Histograms over the number of representations correctly classifying each pair, for matched and mismatched pairs (cut off at 100 pairs).

Figure 4.5 shows all 53 pairs from View 2 that were incorrectly classified by all representations used in our system. These images highlight a fundamental difficulty with face verification, and verification within an object class in general, namely the large amount of intra-class variation due to matched pairs depicting the same individual from different viewpoints, as well as from other nuisance factors such as partial occlusions.

One difficulty specific to LFW is that these matched pairs depicting the same individual from different viewpoints are relatively rare, as the faces had to first be detected by a frontal face detector in order to be included in LFW. Therefore, there may be insufficient training information within LFW itself to properly classify such matched pairs. One solution is to try to add knowledge of how a single face can appear from multiple viewpoints directly into the classification system, such as the approach taken by Yin *et al.* [114], who made use of this information as encoded in the Multi-PIE data set.

It is also interesting to consider less supervised methods of learning this type of three dimensional structure or being more robust to misalignments and occlusions. One possibility is to artificially perturb the training data to introduce such errors; another possibility is to learn correspondences between different viewpoints from video.

4.6 Discussion

We have demonstrated that we can improve upon methods that utilize a combination of representations from hand-crafted image descriptors by adding additional representations from deep learning. We obtain novel representations through a new local convolutional RBM model and by applying deep learning to new visible data such as LBP. By combining such deep learning representations with hand-crafted descriptors, we achieve new state of the art accuracy on the LFW face verification database, and our methodology can be readily applied to other systems as well.

Figure 4.5: All pairs from LFW incorrectly classified by all representations. The four mismatched pairs have a red border; all other pairs are matched pairs.

CHAPTER 5

DEEP LEARNING FOR FACE ALIGNMENT

In Chapter 3, we developed a method for unsupervised joint alignment of images that is able to improve performance on recognition tasks such as face verification. Such alignment removed undesired variability due to factors such as pose, while only requiring weak supervision in the form of poorly aligned examples. However, this work on unsupervised alignment of complex, real world images required the careful selection of feature representation based on hand-crafted image descriptors, in order to achieve an appropriate, smooth optimization landscape.

In this chapter, we instead propose a novel combination of unsupervised joint alignment with the unsupervised feature learning of Chapter 4. Specifically, we incorporate deep learning into the congealing framework. Through deep learning, we obtain features that can capture the image at differing resolution based on network depth, and that is tuned to the statistics of the specific data being aligned. In addition, we modify the learning algorithm for the restricted Boltzmann machine by incorporating a group sparsity penalty, leading to a topographic organization on the learned filters and improving subsequent alignment results.

We apply our proposed algorithm to the unconstrained face images of LFW. Using the aligned images produced by our proposed unsupervised algorithm, we achieve a significantly higher accuracy in face verification than obtained using the original face images, prior work in unsupervised alignment, and prior work in supervised alignment. We also match the accuracy for the best available, but unpublished method.

5.1 Introduction

As previously mentioned, one of the most challenging aspects of image recognition is the large amount of intra-class variability, from factors such as lighting, background, pose, and perspective transformation. For tasks involving a specific object category, such as face verification, this intra-class variability can often be much larger than inter-class differences. Recognition performance can be significantly improved by removing undesired intra-class variability by first aligning the images to some canonical pose or configuration.

For instance, face verification accuracy can be dramatically increased through image alignment, by detecting facial feature points on the image and then warping these points to a canonical configuration. This alignment process can lead to significant gains in recognition accuracy on real world face verification, even for algorithms that were explicitly designed to be robust to some misalignment [112]. Therefore, the majority of face recognition systems evaluated on LFW currently make use of a preprocessed version of the data set known as LFW-a,[1] where the images have been aligned by a commercial fiducial point-based supervised alignment method [104].

Fiducial point (or landmark-based) alignment algorithms [112, 19, 6, 118], however, require a large amount of supervision or manual effort. One must first decide which fiducial points to use for the specific object class, and then obtain many example image patches of these points. These methods are thus hard to apply to new object classes, since all of this manual collection of data must be re-done, and the alignment results may be sensitive to the choice of fiducial points and quality of training examples.

As discussed in Chapter 3, an alternative to this supervised approach is to take a set of poorly aligned images (*e.g.*, images drawn from approximately the same dis-

[1] `http://www.openu.ac.il/home/hassner/data/lfwa/`

tribution as the inputs to the recognition system) and attempt to make the images more similar to each other, using some measure of joint similarity such as entropy. This is the congealing framework, whereby each image in a set of images is iteratively transformed to reduce the total entropy of the set. Earlier, we showed how to extend congealing to work on complex, real-world object classes such as faces and cars. However, this required a careful selection of hand-crafted feature representation (SIFT [61]) and soft clustering, and does not achieve as large of an improvement in verification accuracy as supervised alignment (LFW-a).

In this chapter, we propose a novel combination of unsupervised alignment and unsupervised feature learning by incorporating deep learning [31, 4, 90, 87, 50] into the congealing framework. Through deep learning, we can obtain a feature representation tuned to the statistics of the specific object class we wish to align. Moreover, we can capture the data at multiple scales by using multiple layers of a deep learning architecture. In addition, we incorporate a group sparsity constraint into the deep learning algorithm, leading to a topographic organization on the learned filters, and show that this in turn leads to improved alignment results. We apply our method to unconstrained face images and demonstrate that, using the aligned images, we achieve a significantly higher face verification accuracy than obtained both using the original face images and using the images produced by prior work in unsupervised alignment [35]. In addition, the accuracy surpasses that achieved using supervised fiducial points based alignment [19], and matches the accuracy using the LFW-a images produced by commercial supervised alignment.

5.2 Related Work

Cox *et al.* presented a variation of congealing for unsupervised alignment, where the entropy similarity measure is replaced with a least-squares similarity measure [15, 16]. Liu *et al.* extended congealing by modifying the objective function to allow

for simultaneous alignment and clustering [60]. Zhu *et al.* developed a method for non-rigid alignment using a model parameterized by mesh vertex coordinates in a deformable Lucas-Kanade formulation.

In this chapter, we chose to extend the original congealing method, rather than other alignment frameworks, for several reasons. The algorithm uses entropy as a measure of similarity, rather than variance or least squares, thus allowing for the alignment of data with multiple modes. Unlike other joint alignment procedures [15], the main loop scales linearly with the number of images to be aligned, allowing for a greater number of images to be jointly aligned, smoothing the optimization landscape. Finally, congealing requires only *very weak supervision* in the form of poorly aligned images.

However, our proposed extensions, using features obtained from deep learning, could also be applied to other algorithms, which have only been used with a pixel intensity representation, such as least-squares congealing [15, 16], and [120], which allows for non-rigid transformations but requires additional supervision in the form of object part (*e.g.*, eye) detectors specific to the data to be aligned.

In addition, we augment the learning procedure used to train DBNs by adding a group sparsity term, leading to a set of learned filters with a *linear* topographic organization. This idea is closely related to the Group Lasso for regression [116] and Topographic ICA [40], and has been applied to sparse coding with basis functions that form a generally two-dimensional topological map [46]. We extend this method to basis functions that are learned in a convolutional manner, and to higher-order features obtained from a multi-layer convolutional DBN.

5.3 Methodology

We will be using the congealing terminology as discussed earlier in Section 3.2. We will refer to the congealing algorithm presented previously as SIFT congealing,

in contrast with the congealing variant presented in this section, which we refer to as deep congealing.

Given a set of poorly aligned face images, our goal is to iteratively transform each image to reduce the total entropy over the pooling layer outputs of a CDBN applied to each of the images. For a CDBN with K pooling layer groups, we now have K location stacks at each image location (after max-pooling), over a binary distribution for each location stack.

Given N unaligned face images, let P be the number of pooling units in each group in the top-most layer of the CDBN. We use the pooling unit probabilities, with the interpretation that the pooling unit can be considered as a mixture of sub-units that are on and off [51]. Letting $p_{\alpha,k}^{(n)}$ be the α pooling unit in group k for image n under some transformation U^n, define $D_{\alpha,k}(1) = \frac{1}{N}\sum_{n=1}^{N} p_{\alpha,k}^{(n)}$ and $D_{\alpha,k}(0) = 1 - D_{\alpha,k}(1)$. Then, the entropy for a specific pooling unit is

$$H(D_{\alpha,k}) = -\sum_{s\in\{0,1\}} D_{\alpha,k}(s)\log(D_{\alpha,k}(s)).$$

At each iteration of congealing, we find a transformation for each image that decreases the total entropy $\sum_{k=1}^{K}\sum_{\alpha=1}^{P} H(D_{\alpha,k})$. Note that if $K = 1$, this reduces to the traditional congealing formulation on the binary output of the single pooling layer.

5.3.1 Learning a Topology

As congealing reduces entropy by performing local hill-climbing in the transformation parameters, a key factor in the success of congealing is the smoothness of this optimization landscape. In SIFT congealing, smoothness is achieved through soft clustering and the properties of the SIFT descriptor. Specifically, to compute the descriptor, the gradient is computed at each pixel location and added to a weighted histogram over a fixed number of angles. The histogram bins have a natural circular topology. Therefore, the gradient at each location contributes to two neighboring

histogram bins, weighted using linear interpolation. This leads to a smoother optimization landscape when congealing. For instance, if a face is rotated a fraction of the correct angle to put it into a good alignment, there will be a corresponding partial decrease in entropy due to this interpolated weighting.

In contrast, there is no topology on the filters produced using standard learning of a CRBM. This may lead to plateaus or local minima in the optimization landscape with congealing, for instance, if a section of a face is rotated between two filters. This problem may be particularly severe for filters learned at deeper layers of a CDBN. For instance, a second-layer CRBM trained on face images would likely learn multiple filters that resemble eye detectors, capturing slightly different types and scales of eyes. If these filters are activating independently, then the resulting entropy of a set of images may not decrease even if eyes in different images are brought into closer alignment.

A CRBM is generally trained with sparsity regularization [54], such that each filter responds to a sparse set of input stimuli. A smooth optimization for congealing requires that, as an image patch is transformed from one such sparse set to another, the change in pooling unit activations is also gradual rather than abrupt. Therefore, we would like to learn filters with a linear topological ordering, such that when a particular pooling unit $p_{\alpha,k}$ at location α and associated with filter k is activated, the pooling units at the same location, associated with nearby filters, *i.e.*, $p_{\alpha,k'}$ for k' close to k, will also have partial activation. To learn a topology on the learned filters, we add the following group sparsity penalty to the learning objective function (*i.e.*, negative log-likelihood):

$$\mathcal{L}_{\text{sparsity}} = \lambda \sum_{k,\alpha} \sqrt{\sum_{k'} w_{k'-k} p_{\alpha,k'}^2}$$

where w_d is a Gaussian weighting, $w_d \propto \exp(-\frac{d^2}{2\sigma^2})$.

Let the term *array* be used to refer to the set of pooling units associated with a particular filter, *i.e.*, $p_{\alpha,k}$ for all locations α. This regularization penalty is a sum (L^1 norm) of L^2 norms, each of which is a Gaussian weighting, centered at a particular array, of the pooling units across each array at a specific location. In practice, rather than weighting every array in each summand, we use a fixed kernel covering five consecutive filters, *i.e.*, $w_d = 0$ for $|d| > 2$.

The rationale behind such a regularization term is that an L^1 norm encourages sparsity whereas an L^2 norm does not. This sum of L^2 norms thus encourages sparsity at the group level, where a group is a set of Gaussian weighted activations centered at a particular array. Therefore, if two filters are similar and tend to both activate for the same visible data, then a smaller penalty will be incurred if these filters are nearby in the topological ordering, as this will lead to a more sparse representation at the group L^2 level.

To account for this penalty term, we augment the learning algorithm by taking a step in the negative derivative with respect to the CRBM weights. To compute the derivative, we first need to compute the derivative of the pooling unit with respect to the CRBM weights:

$$
\begin{aligned}
\frac{\partial}{\partial W_{rsk}} p_{\alpha,k} &= \frac{\partial}{\partial W_{rsk}} \frac{\sum_{ij \in B_\alpha} \exp(I(h_{ijk}))}{1 + \sum_{ij \in B_\alpha} \exp(I(h_{ijk}))} \\
&= \frac{\sum_{ij \in B_\alpha} \exp(I(h_{ijk})) \frac{\partial}{\partial W_{rsk}} I(h_{ijk})}{1 + \sum_{ij \in B_\alpha} \exp(I(h_{ijk}))} \\
&\quad - \frac{[\sum_{ij \in B_\alpha} \exp(I(h_{ijk}))][\sum_{ij \in B_\alpha} \exp(I(h_{ijk})) \frac{\partial}{\partial W_{rsk}} I(h_{ijk})]}{(1 + \sum_{ij \in B_\alpha} \exp(I(h_{ijk})))^2} \\
&= \frac{\sum_{ij \in B_\alpha} \exp(I(h_{ijk})) \frac{\partial}{\partial W_{rsk}} I(h_{ijk})}{1 + \sum_{ij \in B_\alpha} \exp(I(h_{ijk}))} \left(1 - \frac{\sum_{ij \in B_\alpha} \exp(I(h_{ijk}))}{1 + \sum_{ij \in B_\alpha} \exp(I(h_{ijk}))}\right) \\
&= \sum_{ij \in B_\alpha} \frac{\exp(I(h_{ijk}))}{1 + \sum_{ij \in B_\alpha} \exp(I(h_{ijk}))} (1 - p_{\alpha,k}) \frac{\partial}{\partial W_{rsk}} I(h_{ijk}) \\
&= \sum_{ij \in B_\alpha} h_{ijk} (1 - p_{\alpha,k}) v_{i+r-1,j+s-1}
\end{aligned}
$$

With this, we can now compute the derivative of the sparsity term as:

$$
\begin{aligned}
\frac{\partial}{\partial W^k_{rs}} \mathcal{L} &= \lambda \sum_{k',\alpha} \frac{1}{2\sqrt{\sum_{k''} w_{k''-k'} p^2_{\alpha,k''}}} \frac{\partial}{\partial W_{rsk}} \left(\sum_{k''} w_{k''-k'} p^2_{\alpha,k''} \right) \\
&= \lambda \sum_{k',\alpha} \frac{1}{\sqrt{\sum_{k''} w_{k''-k'} p^2_{\alpha,k''}}} w_{k-k'} p_{\alpha,k} \frac{\partial}{\partial W_{rsk}} p_{\alpha,k} \\
&= \lambda \sum_{k',\alpha} \frac{1}{\sqrt{\sum_{k''} w_{k''-k'} p^2_{\alpha,k''}}} w_{k-k'} p_{\alpha,k} (1 - p_{\alpha,k}) \sum_{ij \in B_\alpha} h_{ijk} v_{i+r-1,j+s-1} \\
&= \lambda \sum_{k'} \frac{1}{\sqrt{\sum_{k''} w_{k''-k'} p^2_{\alpha,k''}}} w_{k-k'} \sum_{ij} p_{\alpha(ij),k} (1 - p_{\alpha(ij),k}) h_{ijk} v_{i+r-1,j+s-1}
\end{aligned}
$$

If we now define J as

$$
J^k_{ij} = p_{\alpha(ij),k} (1 - p_{\alpha(ij),k}) h_{ijk},
$$

we can efficiently compute the full gradient using convolutions as

$$
\nabla_{W^k} \mathcal{L} = \lambda \sum_{k'} \frac{1}{\sqrt{\sum_{k''} w_{k''-k'} p^2_{\alpha,k''}}} w_{k-k'} (v * \tilde{J}),
$$

where $*$ denotes convolution and $\tilde{J}$ means J flipped horizontally and vertically.

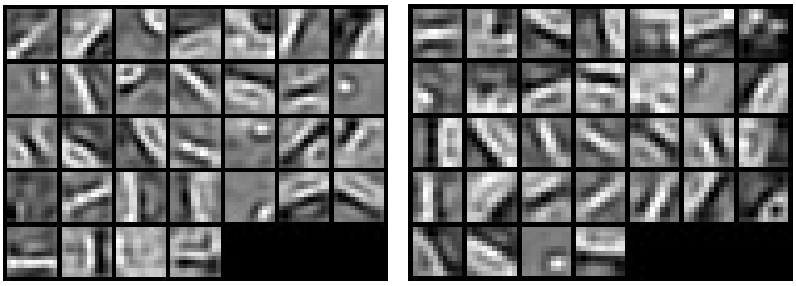
As mentioned in Chapter 4, we initialize the filters using expectation-maximization under a mixture of Gaussians/Bernoullis, before proceedings with convolutional RBM learning. Therefore, when learning with the group sparsity penalty, we periodically reorder the filters using the following greedy strategy. Taking the first filter, we iteratively add filters one by one to the end of the filter set, picking the filter that minimizes the group sparsity penalty.

5.4 Experiments

We learn three different convolutional DBN models to use as the feature representation for deep congealing. First, we learn a one-layer CRBM from the Kyoto images,[2] a standard natural image data set, to evaluate the performance of congealing with self-taught CRBM features. Next, we learn a one-layer CRBM from LFW face images, to compare performance when learning the features directly on images of the object class to be aligned. Finally, we learn a two-layer CRBM from LFW face images, to evaluate performance using higher-order features. For all three models, we also compare learning the weights using the standard sparse CDBN learning, as well as learning with group sparsity regularization. Visualizations of each set of learned weights are given in Figures 5.1, 5.2, and 5.3.

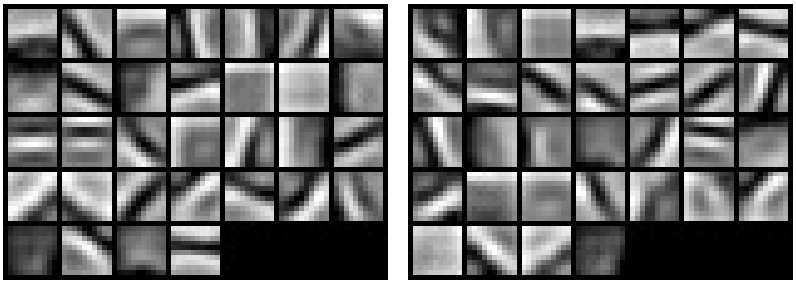
During learning, we used a pooling size of 5x5 for the one-layer models, and a pooling size of 3x3 in both layers of the two-layer model. We used a variance of 1 in the Gaussian weighting for group sparsity regularization. For computing the pooling layer representation to use in congealing, we modified the pooling size to 3x3 for the one-layer models and 2x2 for the second layer in the two-layer model, and adjusted the hidden biases to give an expected activation of 0.025 for the hidden units.

[2] `http://www.cnbc.cmu.edu/cplab/data_kyoto.html`



(a) Without topology (b) With topology

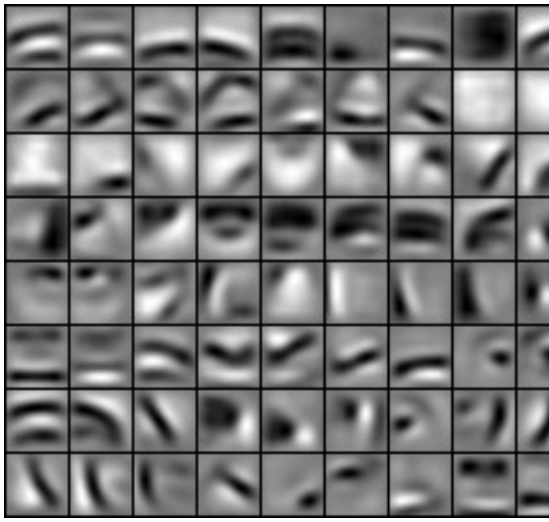
Figure 5.1: Visualization of first layer filters learned from Kyoto natural images, without topology on left and with topology on right. By learning with a linear topology, nearby filters (in row major order) are similar, such as the similarly oriented edge filters in the third and fourth rows, encouraging partial activations in neighboring layers when a pooling unit in a particular layer is activated.



(a) Without topology (b) With topology

Figure 5.2: Visualization of first layer filters learned from face images, without topology on left and with topology on right.

(a) Without topology



(b) With topology

Figure 5.3: Visualization of second layer filters learned from face images, without topology on left and with topology on right. Learning with topology groups together filters for particular facial features, such as eye detectors at the end of the row third from the bottom.

In Figure 5.4, we show a selection of images under several alignment methods, for which the methods produced different transformations. Each image is shown in its original form, and aligned using SIFT Congealing, Deep Congealing with topology, using a one-layer and two-layer CDBN trained on faces, and the LFW-a alignment.

We evaluate the effect of alignment on verification accuracy using View 1 of LFW. For the congealing methods, 400 images from the training set were congealed and used to form a funnel to subsequently align all of the images in both the training and test sets.

As the verification system, we used the SVM LBP classifier presented in Chapter 4. Specifically, we use square root LBP features computed over non-overlapping 10x10 pixel regions from a 150x80 cropped region of the full LFW images. We apply whitening PCA, reducing the representation to 500 dimensions, and normalize the feature vector for each image in a pair before combining using element-wise multipli-

Figure 5.4: Sample images from LFW produced by different alignment algorithms. For each set of five images, the alignments are, from left to right: original images; SIFT Congealing; Deep Congealing, Faces, layer 1, with topology; Deep Congealing, Faces, layer 2, with topology; Supervised (LFW-a).

cation to generate a single feature vector for the pair, which is the input to a linear SVM.

Table 5.1 gives the verification accuracy for this verification system using images produced by a number of alignment algorithms. Using a CDBN representation learned with a group sparsity penalty, leading to learned filters with topographic organization, consistently gives a higher accuracy of one to two percentage points. We compare with two supervised alignment systems, the fiducial points based system of [19],[3] and LFW-a. Note that LFW-a was produced by a commercial alignment system, in the spirit of [19], but with important differences that have not been published [104]. Congealing with a one-layer CDBN trained on faces, with topology, gives verification accuracy significantly higher than using images produced by [19], and comparable to the accuracy using LFW-a images.

Alignment	Accuracy
Original	0.742
SIFT Congealing	0.758
Deep Congealing, Kyoto, layer 1	0.807
Deep Congealing, Kyoto, layer 1, with topology	0.815
Deep Congealing, Faces, layer 1	0.802
Deep Congealing, Faces, layer 1, with topology	0.820
Deep Congealing, Faces, layer 2	0.780
Deep Congealing, Faces, layer 2, with topology	0.797
Combining Scores of Faces, layers 1 and 2, with topology	**0.831**
Fiducial Points-based Alignment [19] (supervised)	0.805
LFW-a (commercial)	0.823

Table 5.1: Unconstrained face verification accuracy on View 1 of LFW using images produced by different alignment algorithms. By combining the classifier scores produced by layer 1 and 2 using a linear SVM, we are able to achieve higher accuracy using unsupervised alignment than obtained using the widely-used LFW-a images, generating using a commercial supervised fiducial-points alignment algorithm.

[3]Using code available at `http://www.robots.ox.ac.uk/~vgg/research/nface/`

Moreover, we can combine the verification scores using images from the one-layer and two-layer CDBN trained on faces, learning a second linear SVM on these verification scores. By doing so, we achieve a further gain in verification performance, achieving an accuracy of 0.831, exceeding the accuracy using LFW-a. This suggests that the two-layer CDBN alignment is somewhat complementary to the one-layer alignment. In other words, although the two-layer CDBN alignment produces a lower verification accuracy, it is not strictly worse than the one-layer CDBN alignment for all images, but rather is aligning according to a different set of statistics, and achieves success on a slightly different subset of images than the one-layer CDBN model. As a control, we performed the same score combination using the scores produced from images from the one-layer CDBN alignment trained on faces, with topology, and the original images. This gave a verification accuracy of 0.817, indicating that the improvement from combining the one and two-layer scores is not merely obtained from using two different sets of alignments.

5.5 Discussion

In this work, we have shown how to combine unsupervised joint alignment with unsupervised feature learning. By congealing on the pooling layer representation given by a CDBN, we are able to achieve significant gains in verification accuracy over existing methods for unsupervised alignment. By adding a group sparsity penalty to the CDBN learning algorithm, we can learn filters with a linear topology, providing a smoother optimization landscape for congealing. Using face images aligned by this method, we obtain higher verification accuracy than the supervised fiducial points based method of [19]. Further, despite being unsupervised, our method is still able to achieve comparable accuracy with the widely used LFW-a images, obtained by a commercial fiducial point-based alignment system whose detailed procedure is not published yet. We thus believe that our proposed method is an important contribution

in developing generic alignment systems that do not require domain-specific fiducial points.

One direction for future work is to optimize the congealing algorithm. In our implementation, one of the main bottlenecks is the time taken to transform each image at each iteration of congealing. Therefore, we limit ourselves to the center 150x150 cropped region of each image. This places a limit on the number of layers and pooling sizes in the CDBNs that can be used. Optimizing the congealing algorithm such that the full 250x250 LFW images can be used will allow for a large number of CDBNs to be used as feature representations, possibly generating better alignments. Another natural extension of this work is to use the local CRBM model presented in Chapter 4 to learn features specific to individual regions of the face and take advantage of the global structure of the face. Our current implementation is however slightly slower than a standard CRBM and would need to be slightly optimized to be used within the congealing algorithm.

CHAPTER 6

CONCLUSIONS

The aim of this dissertation has been to improve face recognition in real-world scenarios where acquisition of the face images cannot be controlled.

First, we developed a data set, LFW, for studying unconstrained face verification that, in contrast to existing face databases at the time, contained face images reflecting the variability encountered in everyday life. Since its introduction, LFW has become a de facto standard for measuring performance on unconstrained face verification, with over 20 published verification methods being evaluated on LFW.

Next, we demonstrated how weakly supervised data, in the form of unlabeled face images, could be used to improve face verification performance. We first show how to extend the unsupervised joint alignment method of congealing to images of complex objects such as faces and cars. By applying congealing to a set of poorly aligned face images, we can automatically align the images and reduce unwanted variation due to pose.

Second, we use deep learning to perform unsupervised feature learning from the unlabeled face images. We develop a new local convolutional restricted Boltzmann machine that takes advantage of global structure by learning filters specific to different regions of the images. We show that we can combine these learned feature representations with standard representations obtained from hand-crafted image descriptors to achieve state of the art face verification results using a single similarity metric.

Lastly, we combine the above two approaches, using deep learning features within a congealing framework. We modify the learning algorithm by adding a sparsity penalty on groups of filters, resulting in a linear topology on the learned filters. By iteratively minimizing the entropy of these filter responses, we are able to perform unsupervised alignment and achieve an improvement in verification accuracy matching that obtained by a supervised alignment based on detecting facial fiducial points.

6.1 Future Work

Although there still exists a significant gap between human and machine performance on LFW, the rate of increase in machine verification performance has recently slowed. This raises a question of whether there is sufficient information within the LFW training data to learn a classifier that achieves near human label performance. As mentioned earlier, the difficult to classify pairs in LFW tend to be matched pairs where the two images are from very different pose angles. In constructing LFW, the primary assumption that was made was that the face images were initially detected by a frontal face detector. Therefore, these difficult to classify pairs are outliers, formed from non-frontal face images at the limit of the face detector's ability to successfully detect.

Since these pairs only form a small fraction of the LFW training data, it may be the case that a classifier could, in principle, learn to properly classify such pairs, but fails to do so due to insufficient training data, as suggested by Pinto and Cox [83]. It is worth noting that the current state of the art method on LFW of Yin *et al.* [114] makes use of outside training data in the form of the Multi-PIE database [28], using information contained in the same faces taken from different views.

One direction for future research would be to create a follow-up to LFW that replaces the frontal face detector with either a multi-view face detector or manually annotated face regions, thereby leading to a broader distribution over face pairs at

differing pose angles. Another possible direction is to learn the type of correspondence between differing views used by Yin *et al*. from weakly supervised data, such as video.

Finally, this dissertation has focused on improving face verification through weakly supervised learning. These techniques leverage more readily obtained training data, so another direction for future work would be to apply these ideas to other verification tasks and finer-grained recognition between objects of the same class. Possible tasks include differentiating between different makes of cars [22] and different types of flowers [74].

APPENDIX A

LFW VIEW 1/VIEW 2 OVERLAP

Rather than a traditional training, validation, and testing split of the data, LFW was organized into two views, View 1 for model selection and View 2 for performance evaluation. As a result, the two views have some overlap. In this appendix, we discuss a consequence of this design choice and the potential for overfitting by improper use of View 1.

A.1 Proper Use of View 1 and View 2

As first indicated in the Labeled Faces in the Wild technical report [38], LFW includes two defined views of the data: View 1 for model selection and algorithm development, and View 2 for performance reporting. The rationale for allowing data reuse between views, making the views not mutually exclusive, was to allow for larger training and test set sizes. As stated in Chapter 2, although this leads to some bias in the results, we argue that this bias should be small and outweighed by the benefit of larger set sizes.

Due to this overlap, however, we cautioned against training methods that may inadvertently memorize instances from View 1. It is important to note that these views do not form a traditional training/validation/testing split of the data; in particular, performance is measured on ten separate folds of View 2, each with its own defined training data. To draw attention to this issue, we here give examples of two methods that may have potentially inadvertently overfitted to the test data due to inappropriate usage of View 1 data.

In their paper, Pinto and Cox [82] mistakenly assume that View 1 and View 2 are mutually exclusive,[1] emphasizing that this allows them to tune performance on View 1 while avoiding selection bias artifacts. As we highlight later, there is in fact a large overlap between View 1 and View 2, creating the potential for inadvertent but significant over-fitting to the test data. As they perform brute force search using clusters of high-end graphics hardware, we are unable to re-implement their method and train solely on View 2, and thus cannot directly test to what extent their method benefited from their use of View 1 data.

From personal communication with Nguyen and Bai [71], we learned that the performance accuracy published in their paper used a different training strategy than presented in the paper. To achieve the results in the paper, they performed cosine similarity metric learning using View 1 training as the training set, and View 1 testing as the validation set. They then applied the learned metric to View 2, only using the View 2 training data for each fold to adjust the threshold for determining matched and mismatched pairs.

Although they no longer have saved results using the training strategy outlined in the paper, which only made use of View 2, we were able to run a comparison using our own implementation of their algorithm.

Following the training strategy as presented in the paper (in our view, the proper strategy in accordance with the intended use), their system consistently performed worse than the published results, obtaining 81.8% accuracy using the square root LBP feature representation. By improperly training on View 1 rather than View 2, we increased our accuracy to 83.3%, a statistically significant increase, despite the fact that View 1 has significantly less available training data than using 9 folds of View 2, strongly suggesting that overfitting is occurring.

---

[1] From the paper: "Note that LFW View 1 and View 2 do not contain the same individuals and are thus mutually exclusive sets."

Category	Number	In View 1	%
View 1 Images	4491		
View 1 Pairs	3200		
View 2 Images	7701	3637	47.2
View 2 Pairs	6000	758	12.6

View 2	Type Overlap	In View 1	%
Matched Pairs	Exact	758	12.6
Matched Pairs	Both Images	244	8.1
Matched Pairs	One Image	927	30.9
Mismatched Pairs	Exact	0	0
Mismatched Pairs	Both Images	681	22.7
Mismatched Pairs	One Image	1488	49.6
Pairs	Any	4098	68.3

Table A.1: Top: Number of unique images appearing in at least one pair, and number of pairs, in both views of LFW; and subset of View 2 also present in View 1. Bottom: For pairs in View 2, the degree to which the pair is present in View 1 as well, *e.g.*, "Both Images" means that both images in the pair are present in View 1, but not together as a pair.

A.1.1 Overlap in Views

To get a better sense of the amount of overlap between View 1 and View 2, and hence potential for overfitting to the test data, we generated some statistics presented in Table A.1. Out of the total number of unique images appearing in at least one pair of View 2, nearly half also appear in at least one pair of View 1. Out of the 6000 pairs used in View 2, 758 also appear in View 1. Moreover, for 4098 of the pairs in View 2, at least one image in the pair appears in View 1.

Following the suggested use recommendation in the LFW technical report, and using View 1 to set a small number of hyper-parameters, such as choice of kernel or number of features, probably does not lead to much overfitting, despite this overlap. However, using View 1 to learn many parameters of a high-capacity system, such as the specific feature representation in high-throughput feature learning or the learned metric in cosine-similarity metric learning, has the potential to be unfairly overfitting

to the test data, given that 12.6% of the test pairs are seen at training time, and for an additional 60% of the test pairs, at least one of the images in the pair is seen at training time, allowing the system to learn how to either match that image to one of its true matches, or discriminate it from at least one false match.

Between these extremes of setting a small number of hyper-parameters to setting large numbers of parameters in a high-capacity system, there is a continuum of possible training strategies. Given the lack of a clear threshold indicating how much use of View 1 is acceptable in setting parameters without significantly benefiting from the overlap with View 2, this problem suggests that in the long run, an ideal data set should contain sufficiently many examples to allow for mutually exclusive training, validation, and test sets.

APPENDIX B

MIXTURE OF BERNOULLIS

Recently, Sohn *et al.* [101] proposed an efficient training algorithm for sparse, convolutional RBMs by establishing connections between Gaussian mixture models and sparse Gaussian RBMs. In this appendix, we provide the mathematical details for extending this efficient training algorithm to learning sparse binary convolutional RBMs, used in the second and higher layers of a deep network.

To do so, we show an equivalence between Bernoulli mixture models and binary RBMs with a softmax constraint, enabling direct conversion from one model to the other. The softmax constraint can then be relaxed into the sparse RBM of Lee *et al.* [54] in the same manner as in Sohn *et al.* Using this training method, sparse RBMs can be learned with almost no hyperparameter tuning.

B.1 Bernoulli Mixture Models

A Bernoulli mixture model with observed variables $\{v_i\}$ and hidden variables $\{h_j\}$ indicating mixture component can be defined as

$$\begin{aligned} P(h = j) &= \pi_j \\ P(v_i = 1|h = j) &= \sigma(W_{ij} + c_i), \end{aligned}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function, π_j is the prior probability of mixture component j, and W_{ij} and c_j are the parameters for mixture component j.

The joint probability can thus be computed as

$$P(v, h = j) = \pi_j \prod_{i=1}^{d} \left[\sigma(W_{ij} + c_i)^{v_i} (1 - \sigma(W_{ij} + c_i))^{1-v_i} \right],$$

and the posterior probability can be computed as

$$P(h = j|v) = \frac{\pi_j \prod_{i=1}^{d} \left[\sigma(W_{ij} + c_i)^{v_i} (1 - \sigma(W_{ij} + c_i))^{1-v_i} \right]}{\sum_{j'} \pi_{j'} \prod_{i=1}^{d} \left[\sigma(W_{ij'} + c_i)^{v_i} (1 - \sigma(W_{ij'} + c_i))^{1-v_i} \right]}$$

This model can be trained with the Expectation Maximization (EM) algorithm.

B.1.1 Binary RBM with a Softmax Constraint

The binary RBM (with softmax constraint) can be written as

$$\begin{aligned} P(v, h) &= \frac{1}{Z} \exp(v^T W h + b^T h + c^T v) \\ \text{subj. to} & \quad \sum_j h_j \leq 1. \end{aligned}$$

The partition function Z can be written as

$$\begin{aligned} Z &= \sum_{j=1}^{N} \sum_{v} \exp(\sum_{i} (W_{ij} + c_i) v_i + b_j) \\ &= \sum_{j=1}^{N} \exp(b_j) \prod_{i=1}^{d} \left[1 + \exp(W_{ij} + c_i) \right]. \end{aligned}$$

We can first verify that the conditional probability $P(v|h = j)$ is the same as that of the Bernoulli mixture model:

$$\begin{aligned} P(v|h = j) &= \frac{\exp(v^T W_j + b_j + c^T v)}{\exp(b_j) \prod_{i=1}^{d} \left[1 + \exp(W_{ij} + c_i) \right]} \\ &= \prod_{i=1}^{d} \sigma(W_{ij} + c_i)^{v_i} \left(1 - \sigma(W_{ij} + c_i) \right)^{1-v_i}. \end{aligned}$$

We then write the prior $P(h)$ and match it with that of the Bernoulli mixture model:

$$\begin{aligned} P(h=j) &= \frac{1}{Z}\exp(b_j)\prod_{i=1}^{d}\left[1+\exp(W_{ij}+c_i)\right] \\ &= \frac{\exp(b_j)\prod_{i=1}^{d}\left[1+\exp(W_{ij}+c_i)\right]}{\sum_{j'=1}^{N}\exp(b_{j'})\prod_{i=1}^{d}\left[1+\exp(W_{ij'}+c_i)\right]} \\ &= \pi_j. \end{aligned}$$

Solving for b_j, we obtain

$$b_j = \log\pi_j - \sum_{i=1}^{d}\log\left[1+\exp(W_{ij})\right] + \log k.$$

The constant k can be canceled out when normalizing with the partition function, so we can simply write b_j as

$$b_j = \log\pi_j - \sum_{i=1}^{d}\log\left[1+\exp(W_{ij})\right].$$

We have now established the conversion formula between the Bernoulli mixture model and binary RBM with softmax constraint. We can also verify that the posterior probability under the binary RBM is equivalent to the posterior probability under the Bernoulli mixture model:

$$\begin{aligned} P(h=j|v) &= \frac{\exp(v^TW_j+b_j+c^Tv)}{\sum_{j'}\exp(v^TW_{j'}+b_{j'}+c^Tv)} \\ &= \frac{\exp(b_j)\prod_{i=1}^{d}\exp(v_iW_{ij}+v_ic_i)}{\sum_{j'}\exp(b_{j'})\prod_{i=1}^{d}\exp(v_iW_{ij'}+v_ic_i)} \\ &= \frac{\pi_j\frac{\prod_{i=1}^{d}\exp(v_iW_{ij}+v_ic_i)}{\prod_{i=1}^{d}\log[1+\exp(W_{ij}+v_ic_i)]}}{\sum_{j'}\pi_{j'}\frac{\prod_{i=1}^{d}\exp(v_iW_{ij'}+v_ic_i)}{\prod_{i=1}^{d}\log\left[1+\exp(W_{ij'}+v_ic_i)\right]}} \\ &= \frac{\pi_j\prod_{i=1}^{d}\sigma(W_{ij}+c_i)^{v_i}\left[1-\sigma(W_{ij}+c_i)\right]^{1-v_i}}{\sum_{j'}\pi_{j'}\prod_{i=1}^{d}\sigma(W_{ij'}+c_i)^{v_i}\left[1-\sigma(W_{ij'}+c_i)\right]^{1-v_i}}. \end{aligned}$$

B.1.2 Conversion from Bernoulli Mixture Model to Binary RBM

Based on the equivalence shown above, we can convert from the Bernoulli mixture model to binary RBM with softmax constraint using the following formulas:

$$\begin{aligned} W_{ij} &= W_{ij} \\ b_j &= \log \pi_j - \sum_{i=1}^{d} \log \left[1 + \exp(W_{ij})\right]. \end{aligned}$$

Training a Bernoulli mixture model via EM is significantly easier than learning a sparse binary RBM. Therefore, by first learning a Bernoulli mixture model, one can obtain a good initialization to begin training a sparse binary RBM.

BIBLIOGRAPHY

[1] Angelova, Anelia, Abu-Mostafa, Yaser, and Perona, Pietro. Pruning training sets for learning of object categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2005), pp. 495–501.

[2] Arora, Himanshu, Loeff, Nicolas, Forsyth, David, and Ahuja, Narendra. Unsupervised segmentation of objects using efficient learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2007).

[3] Belhumeur, Peter N., Hespanha, Joao, and Kriegman, David J. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1997).

[4] Bengio, Yoshua, Lamblin, Pascal, Popovici, Dan, and Larochelle, Hugo. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems* (2007).

[5] Berg, Tamara L., Berg, Alexander C., Edwards, Jaety, and Forsyth, David A. Who's in the picture. In *Advances in Neural Information Processing Systems* (2004).

[6] Berg, Tamara L., Berg, Alexander C., Maire, Michael, White, Ryan, Teh, Yee Whye, Learned-Miller, Erik, and Forsyth, David A. Names and faces in the news. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2004).

[7] Beveridge, J. Ross, Griffith, Joey, Kohler, Ralf, Hanson, Allen, and Riseman, Edward. Segmenting images using localized histograms and region merging. *International Journal of Computer Vision 2*, 3 (1989).

[8] Beymer, David, and Poggio, Tomaso. Face recognition from one example view. Tech. Rep. AIM-1536, MIT Artificial Intelligence Laboratory, 1995.

[9] Boureau, Y-Lan, Bach, Francis R., LeCun, Yann, and Ponce, Jean. Learning mid-level features for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2010).

[10] Canon. Technology used in compact digital cameras. `http://www.canon.com/technology/canon_tech/explanation/dc.html`. Accessed: 05/01/2012.

[11] Cao, Zhimin, Yin, Qi, Tang, Xiaoou, and Sun, Jian. Face recognition with learning-based descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2010).

[12] Chapelle, O., Schölkopf, B., and Zien, A. *Semi-supervised learning*. MIT Press, 2006.

[13] Collobert, Ronan, and Weston, Jason. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning* (2008).

[14] Cootes, Timothy F., Edwards, Gareth J., and Taylor, Christopher J. Active appearance models. In *Proceedings of the European Conference on Computer Vision* (1998).

[15] Cox, Mark, Lucey, Simon, Sridharan, Sridha, and Cohn, Jeffrey. Least squares congealing for unsupervised alignment of images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2008).

[16] Cox, Mark, Sridharan, Sridha, Lucey, Simon, and Cohn, Jeffrey. Least-Squares congealing for large numbers of images. In *Proceedings of the International Conference on Computer Vision* (2009).

[17] Davis, Jason V., Kulis, Brian, Jain, Prateek, Sra, Suvrit, and Dhillon, Inderjit S. Information-theoretic metric learning. In *Proceedings of the International Conference on Machine Learning* (2007).

[18] Doi, Eizaburo, Inui, Toshio, Lee, Te-Won, Wachtler, Thomas, and Sejnowski, Terrence J. Spatiochromatic receptive field properties derived from information-theoretic analyses of cone mosaic responses to natural scenes. *Neural Computation 15* (2003), 397–417.

[19] Everingham, Mark, Sivic, Josef, and Zisserman, Andrew. "Hello! My name is... Buffy" - automatic naming of characters in TV video. In *Proceedings of British Machine Vision Conference* (2006).

[20] Felzenszwalb, Pedro, and Huttenlocher, Daniel. Efficient graph-based image segmentation. *International Journal of Computer Vision 59*, 2 (2004).

[21] Ferencz, Andras, Learned-Miller, Erik, and Malik, Jitendra. Building a classification cascade for visual identification from one example. In *Proceedings of the International Conference on Computer Vision* (2005).

[22] Ferencz, Andras, Learned-Miller, Erik, and Malik, Jitendra. Learning hyperfeatures for visual identification. In *Advances in Neural Information Processing Systems* (2005), vol. 17, pp. 425–432.

[23] Gardham, Duncan. Airport face scanners 'cannot tell the difference between Osama bin Laden and Winona Ryder'. http://www.telegraph.co.uk/news/uknews/law-and-order/5110402/Airport-face-scanners-cannot-tell-the-difference-between-Osama-bin-Laden-and-Winona-Ryder.html, The Telegraph, 2009. Accessed: 05/01/2012.

[24] Gehler, Peter, and Nowozin, Sebastian. On feature combination for multiclass object classification. In *Proceedings of the International Conference on Computer Vision* (2009).

[25] Georghiades, Athinodoros S., Belhumeur, Peter N., and Kriegman, David J. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 6 (2001), 643–660.

[26] Geuss, Megan, and Purewal, Sarah Jacobsson. Facebooks facial recognition flops. http://www.pcworld.com/article/230318/facebooks_facial_recognition_flops.html, PCWorld, 2011. Accessed: 05/01/2012.

[27] Goodfellow, Ian J., Le, Quoc V., Saxe, Andrew M., Lee, Honglak, and Ng, Andrew Y. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems* (2009), vol. 22.

[28] Gross, Ralph, Matthews, Iain, Cohn, Jeffrey, Kanade, Takeo, and Baker, Simon. Multi-PIE. In *Proceedings of the Eighth IEEE International Conference on Automatic Face and Gesture Recognition* (2008).

[29] Guillaumin, Matthieu, Verbeek, Jakob, and Schmid, Cordelia. Is that you? Metric learning approaches for face identification. In *Proceedings of the International Conference on Computer Vision* (2009).

[30] Hinton, Geffrey E. Training products of experts by minimizing contrastive divergence. *Neural Computation 14*, 8 (2002), 1771–1800.

[31] Hinton, Geoffrey E., Osindero, Simon, and Teh, Yee-Whye. A fast learning algorithm for deep belief nets. *Neural Computation 18*, 7 (2006), 1527–1554.

[32] Hoiem, Derek, Efros, Alexei, and Hebert, Martial. Recovering surface layout from an image. *International Journal of Computer Vision 75*, 1 (2007).

[33] Holub, Alex, Welling, Max, and Perona, Pietro. Combining generative models and fisher kernels for object recognition. In *Proceedings of the International Conference on Computer Vision* (2005).

[34] Hu, Changbo, Feris, Rogerio, and Turk, Matthew. Real-time view-based face alignment using active wavelet networks. In *International Workshop on Analysis and Modeling of Faces and Gestures* (2003).

[35] Huang, Gary B., Jain, Vidit, and Learned-Miller, Erik. Unsupervised joint alignment of complex images. In *Proceedings of the International Conference on Computer Vision* (2007).

[36] Huang, Gary B., Jones, Michael J., and Learned-Miller, Erik. LFW results using a combined Nowak plus MERL recognizer. In *Faces in Real-Life Images Workshop in European Conference on Computer Vision* (2008).

[37] Huang, Gary B., Narayana, Manjunath, and Learned-Miller, Erik. Towards unconstrained face recognition. In *Sixth IEEE Computer Society Workshop on Perceptual Organization in Computer Vision IEEE CVPR* (2008).

[38] Huang, Gary B., Ramesh, Manu, Berg, Tamara, and Learned-Miller, Erik. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[39] Huang, Yuchi, Lin, Stephen, Li, Stan Z., Lu, Hanqing, and Shum, Heung-Yeung. Face alignment under variable illumination. In *Proceedings of 6th IEEE International Conference on Automatic Face and Gesture Recognition* (2004).

[40] Hyvärinen, Aapo, Hoyer, Patrik O., and Inki, Mika. Topographic independent component analysis. *Neural Computation 13*, 7 (2001), 1527–1558.

[41] Jain, Vidit, Ferencz, Andras, and Learned-Miller, Erik. Discriminative training of hyper-feature models for object identification. In *Proceedings of British Machine Vision Conference* (2006).

[42] Jain, Vidit, and Learned-Miller, Erik. FDDB: A benchmark for face detection in unconstrained settings. Tech. Rep. 10-009, University of Massachusetts, Amherst, 2010.

[43] Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. What is the best multi-stage architecture for object recognition? In *Proceedings of the International Conference on Computer Vision* (2009).

[44] Jesorsky, O., Kirchberg, K., and Frischolz, R. Robust face detection using the Hausdorff distance. In *Audio and Video Based Person Authentication*, J. Bigun and F. Smeraldi, Eds. Springer, 2001, pp. 90–95.

[45] Jones, Michael, and Viola, Paul. Fast multi-view face detection. Tech. Rep. TR2003-096, Mitsubishi Electric Research Laboratories Technical Report, 2003.

[46] Kavukcuoglu, Koray, Ranzato, Marc'Aurelio, Fergus, Rob, and LeCun, Yann. Learning invariant features through topographic filter maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[47] Kim, Junhwan, Kolmogorov, Vladimir, and Zabih, Ramin. Visual correspondence using energy minimization and mutual information. In *Proceedings of the International Conference on Computer Vision* (2003).

[48] Kliper-Gross, Orit, Hassner, Tal, and Wolf, Lior. The Action Similarity Labeling Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2012).

[49] Kumar, Neeraj, Berg, Alexander C., Belhumeur, Peter N., and Nayar, Shree K. Attribute and simile classifiers for face verification. In *Proceedings of the International Conference on Computer Vision* (2009).

[50] Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the International Conference on Machine Learning* (2007).

[51] Learned-Miller, Erik. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2005).

[52] Learned-Miller, Erik, and Jain, Vidit. Many heads are better than one: Jointly removing bias from multiple MRIs using nonparametric maximum likelihood. In *Proceedings of Information Processing in Medical Imaging* (2005), pp. 615–626.

[53] Lee, H., Battle, A., Raina, R., and Ng, A. Y. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems* (2007).

[54] Lee, Honglak, Ekanadham, Chaitu, and Ng, Andrew Y. Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems* (2008), vol. 20.

[55] Lee, Honglak, Grosse, Roger, Ranganath, Rajesh, and Ng, Andrew Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the International Conference on Machine Learning* (2009).

[56] Lee, Honglak, Largman, Yan, Pham, Peter, and Ng, Andrew Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems* (2009), vol. 22.

[57] Li, Fei-Fei, Fergus, Rob, and Perona, Pietro. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *Workshop on Generative-Model Based Vision at CVPR* (2004).

[58] Li, Peng, Fu, Yun, Mohammed, Umar, Elder, James H., and Prince, Simon J.D. Probabilistic models for inference about identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 1 (January 2012), 144–157.

[59] Li, Stan Z., Shuicheng, Yan, Zhang, Hongjiang, and Cheng, Qiansheng. Multi-view face alignment using direct appearance models. In *Proceedings of 5th IEEE International Conference on Automatic Face and Gesture Recognition* (2002).

[60] Liu, Xiaoming, Tong, Yan, and Wheeler, Frederick W. Simultaneous alignment and clustering for an image ensemble. In *Proceedings of the International Conference on Computer Vision* (2009).

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

[62] Mattar, Marwan, Ross, Michael, and Learned-Miller, Erik. Nonparametric curve alignment. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (2009).

[63] Miller, Erik, Matsakis, Nick, and Viola, Paul. Learning from one example through shared densities on transforms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2000).

[64] Miller, Erik G. *Learning from one example in machine vision by sharing probability densities*. PhD thesis, Massachusetts Institute of Technology, 2002.

[65] Moghaddam, Baback, Jebara, Tony, and Pentland, Alex. Bayesian face recognition. *Pattern Recognition* (2002).

[66] Mohamed, Abdel Rahman, Dahl, George, and Hinton, Geoffrey E. Deep belief networks for phone recognition. In *Proceedings of NIPS 2009 Workshop on Deep Learning for Speech Recognition and Related Applications* (2009).

[67] Mori, Greg. Guiding model search using segmentation. In *Proceedings of the International Conference on Computer Vision* (2005).

[68] Murphy, Shelley, and Bray, Hiawatha. Face recognition devices failed in test at Logan. `http://www.boston.com/news/local/articles/2003/09/03/face_recognition_devices_failed_in_test_at_logan/`, The Boston Globe, 2003. Accessed: 05/01/2012.

[69] Nair, Vinod, and Hinton, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning* (2010).

[70] National Institute of Standards and Technology. The Color FERET Database. http://www.itl.nist.gov/iad/humanid/colorferet/home.html, 2003.

[71] Nguyen, Hieu V., and Bai, Li. Cosine similarity metric learning for face verification. In *Proceedings of the Asian Conference on Computer Vision* (2010).

[72] Nigam, Kamal, McCallum, Andrew Kachites, Thrun, Sebastian, and Mitchell, Tom. Text classification from labeled and unlabeled documents using EM. *Machine Learning 39* (2000), 103–134.

[73] Nikon. Face-priority AF. `http://www.nikon.com/news/2005/0216_06.htm`. Accessed: 05/01/2012.

[74] Nilsback, Maria-Elena, and Zisserman, Andrew. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing* (2008).

[75] Nowak, Eric, and Jurie, Frédéric. Learning visual similarity measures for comparing never seen objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2007).

[76] Ojala, Timo, Pietikinen, Matti, and Harwood, David. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition 19*, 3 (1996), 51–59.

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

[78] Ozkan, Derya, and Duygulu, Pinar. A graph based approach for naming faces in news photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2006).

[79] Payton, Mark, Greenstone, Matthew, and Schenker, Nathaniel. Overlapping confidence intervals or standard error intervals: What do they mean in terms of statistical significance? *Journal of Insect Science 3*, 34 (2003).

[80] Phillips, P. Jonathon, Flynn, Patrick J., Scruggs, Todd, Bowyer, Kevin, Chang, Jin, Hoffman, Kevin, Marques, Joe, Min, Jaesik, and Worek, William. Overview of the Face Recognition Grand Challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2005).

[81] Phillips, P. Jonathon, Moon, Hyeonjoon, Rizvi, Syed A., and Rauss, Patrick J. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 10 (October 2000), 1090–1104.

[82] Pinto, Nicolas, Barhomi, Youssef, Cox, David D., and DiCarlo, James J. Comparing state-of-the-art visual features on invariant object recognition tasks. In *IEEE Workshop on Applications of Computer Vision* (2011).

[83] Pinto, Nicolas, and Cox, David D. Beyond simple features: A large-scale feature search approach to unconstrained face recognition. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition* (2011).

[84] Pinto, Nicolas, DiCarlo, James J., and Cox, David D. Establishing good benchmarks and baselines for face recognition. In *Faces in Real-Life Images Workshop at ECCV* (2008).

[85] Pinto, Nicolas, DiCarlo, James J., and Cox, David D. How far can you get with a modern face recognition test set using only simple features? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[86] Raina, Rajat, Battle, Alexis, Lee, Honglak, Packer, Benjamin, and Ng, Andrew Y. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the International Conference on Machine Learning* (2007).

[87] Ranzato, Marc'Aurelio, Boureau, Y-Lan, and LeCun, Yann. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems* (2007).

[88] Ranzato, Marc'Aurelio, and Hinton, Geoffrey E. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2010).

[89] Ranzato, Marc'Aurelio, Huang, Fu-Jie, Boureau, Y-Lan, and LeCun, Yann. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2007).

[90] Ranzato, Marc'Aurelio, Poultney, Christopher, Chopra, Sumit, and LeCun, Yann. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems* (2006), pp. 1137–1144.

[91] Ranzato, Marc'Aurelio, Susskind, Joshua, Mnih, Volodymyr, and Hinton, Geoffrey. On deep generative models with applications to recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2011).

[92] Ren, Xiaofeng, and Malik, Jitendra. Learning a classification model for segmentation. In *Proceedings of the International Conference on Computer Vision* (2003).

[93] Ricoh. Face detection technology. `http://www.ricoh.com/about/company/technology/tech/016.html`. Accessed: 05/01/2012.

[94] Roth, Stefan, and Black, Michael J. Fields of experts. *International Journal of Computer Vision 82*, 2 (April 2009), 205–229.

[95] Rowley, Henry, Baluja, Shumeet, and Kanade, Takeo. Rotation invariant neural network-based face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (1998).

[96] Salakhutdinov, Ruslan, and Hinton, Geoffrey E. Semantic hashing. *International Journal of Approximate Reasoning 50* (2009), 969–978.

[97] Sanderson, Conrad, and Lovell, Brian C. Multi-region probabilistic histograms for robust and scalable identity inference. In *International Conference on Biometrics* (2009).

[98] Saxe, Andrew, Koh, Pang Wei, Chen, Zhenghao, Bhand, Maneesh, Suresh, Bipin, and Ng, Andrew. On random weights and unsupervised feature learning. In *Deep Learning and Unsupervised Feature Learning Workshop at NIPS* (2010).

[99] Seo, Hae Jong, and Milanfar, Peyman. Face verification using the LARK representation. *IEEE Transactions on Information Forensics and Security* (2011).

[100] Sim, Terence, Baker, Simon, and Bsat, Maan. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 12 (2003), 1615–1618.

[101] Sohn, Kihyuk, Jung, Dae Yon, Lee, Honglak, and III, Alfred Hero. Efficient learning of sparse, distributed, convolutional feature representations for object recognition. In *Proceedings of the International Conference on Computer Vision* (2011).

[102] Sonnenburg, Soeren, Raetsch, Gunnar, Henschel, Sebastian, Widmer, Christian, Behr, Jonas, Zien, Alexander, de Bona, Fabio, Binder, Alexander, Gehl, Christian, and Franc, Vojtech. The SHOGUN machine learning toolbox. *Journal of Machine Learning Research 11* (June 2010), 1799–1802.

[103] Susskind, Joshua, Memisevic, Roland, Hinton, Geoffrey E., and Pollefeys, Marc. Modeling the joint density of two images under a variety of transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2011).

[104] Taigman, Yaniv, Wolf, Lior, and Hassner, Tal. Multiple one-shots for utilizing class label information. In *Proceedings of British Machine Vision Conference* (2009).

[105] Thrun, Sebastian. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems* (1996).

[106] Turk, Matthew, and Pentland, Alex. Face recognition using Eigenfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (1991).

[107] Viola, Paul, and Jones, Michael. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2001).

[108] Viola, Paul, and Jones, Michael. Robust real-time face detection. *International Journal of Computer Vision* (2004).

[109] Wang, Peng, Tran, Lam Cam, and Ji, Qiang. Improving face recognition by online image alignment. *Pattern Recognition* (2006).

[110] Welling, Max, Hinton, Geoffrey E., and Osindero, Simon. Learning sparse topographic representations with products of student-t distributions. In *Advances in Neural Information Processing Systems* (2003), vol. 15.

[111] Wolf, Lior, Hassner, Tal, and Taigman, Yaniv. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop at ECCV* (2008).

[112] Wolf, Lior, Hassner, Tal, and Taigman, Yaniv. Similarity scores based on background samples. In *Proceedings of the Asian Conference on Computer Vision* (2009).

[113] Yang, Jianchao, Yu, Kai, Gong, Yihong, and Huang, Thomas S. Linear spatial pyramid matching using sparse coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 1794–1801.

[114] Yin, Qi, Tang, Xiaoou, and Sun, Jian. An associate-predict model for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2011).

[115] Ying, Yiming, and Li, Peng. Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research (Special Topics on Kernel and Metric Learning)* (2012).

[116] Yuan, Ming, and Yin, Li. Model selection and estimation in regression with grouped variables. Tech. rep., University of Wisconsin, 2004.

[117] Zeiler, Matthew, Krishnan, Dilip, Taylor, Graham, and Fergus, Rob. Deconvolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2010).

[118] Zhou, Yi, Gu, Lie, and Zhang, Hong-Jiang. Bayesian tangent shape model: Estimating shape and pose parameters via Bayesian inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2003).

[119] Zhou, Yi, Zhang, Wei, Tang, Xiaoou, and Shum, Harry. A Bayesian mixture model for multi-view face alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2005).

[120] Zhu, Jianke, Gool, Luc Van, and Hoi, Steven C.H. Unsupervised face alignment by nonrigid mapping. In *Proceedings of the International Conference on Computer Vision* (2009).

[121] Zollei, Lilla, Learned-Miller, Erik, Grimson, Eric, and Wells, William. Efficient population registration of 3d data. In *Workshop on Computer Vision for Biomedical Image Applications: Current Techniques and Future Trends, at ICCV* (2005).